Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

Cluster generation using tabu search based maximum descent algorithm

J. S. Pan[1], S. C. Chu[1,2] & Z. M. Lu[3]
[1] *National Kaohsiung Institute of Technology, Taiwan*
[2] *University of South Australia, Australia*
[3] *Harbin Institute of Technology, Harbin, China*

Abstract

The maximum descent (MD) algorithms have been proposed for clustering objects. Compared with the traditional K-means (or GLA, or LBG) algorithm, the MD algorithms achieve better clustering performance with far less computation time. However, the searching of the optimal partitioning hyperplane of a multidimensional cluster is a difficult problem in the MD algorithms. In this paper, a new partition technique based on tabu search (TS) approach is presented for the MD algorithms. Experimental results show that the tabu search based MD algorithm can produce a better clustering performance than the K-means and MD algorithms.

1 Introduction

Clustering plays an important role in data mining, pattern recognition and data compression. One of the main applications in clustering techniques is the vector quantization [1]. The k-dimensional, N-level vector quantizer is defined as a mapping from a k-dimensional Euclidean space R^k into a certain finite set C of R^k. This finite set $C=\{Y_1,Y_2,\ldots,Y_N\}$ is called the centers of the clusters (or VQ codebook). Each $Y_i \in R^k$ in clusters (or codebook) C is called center of cluster (or codeword), i=1,2,...,N. The squared Euclidean distortion measure is often used to measure the distortion between the input vector $X=\{x_1,x_2,\ldots,x_k\}$ and the codeword $Y_i=\{y_{i1},y_{i2},\ldots,y_{ik}\}$, i.e.,

$$d(X,Y_i) = \sum_{l=1}^{k}(x_l - y_{il})^2 \tag{1}$$

The key problem of VQ is to generate a good codebook from a number of training vectors. The codebook design problem is essentially a clustering problem that clusters a training set $S=\{X_1,X_2,\ldots,X_M\}$ into N subsets $S_1, S_2,\ldots, S_N$,

which satisfy $$\bigcup_{i=1}^{N} S_i = S \tag{2}$$

and $$S_i \cap S_j = \Phi \text{ if } i \neq j \tag{3}$$

The centriod of the subset S_i is the codeword Y_i, and if

$$d(X_l,Y_i) = \min_{1\le j\le N} d(X_l,Y_j) \tag{4}$$

then $X_l \in S_i$. The aim of the codebook design algorithm is to minimize the total distortion defined as follow:

$$D = \sum_{l=1}^{M} \min_{1\le j\le N} d(X_l,Y_j) \tag{5}$$

A generalized Lloyd clustering algorithm [2] denoted as LBG (or K-means, or GLA) algorithm, is typically used to generate VQ codebooks. However, this iterative algorithm depends on the initial codebook, often obtains the local optimal codebook and needs intensive computation. Thus the two prominent problems in codebook design are how to approach the global optimum and how to reduce the computational complexity. In order generate better codebooks while reduce the computation time, a maximum descent (MD) algorithm [3] has been proposed for VQ codebook design. This algorithm begins with treating the training vector set $S=\{X_1,X_2,\ldots,X_M\}$ as a global cluster. And then the algorithm generates the required number of clusters one by one subject to the maximum distortion reduction criterion until the desired number of codewords are obtained. Compared with the LBG algorithm, the codebook performance is improved and the computational time is substantially reduced. However, the searching of the optimal partitioning hyperplane of a multidimensional cluster is a difficult problem in the MD algorithm. Three techniques [4] have been presented to search the optimal partitioning hyperplane for the MD method. However, these techniques all restrict the searching range among hyperplanes that are perpendicular to the basis vectors of the vector space that can be obtained using discrete cosine transform (DCT), so they can hardly find the global optimal partitioning hyperplane. In order to find the global or nearly global optimal partitioning hyperplane, the tabu search (TS) approach first proposed by Glover [5] is introduced in this paper. The main problem in MD is how to split one cluster into two clusters with minimal mean squared error, it is a combinatorial optimization problem, so the tabu search approach is suitable to solve this problem.

2 The maximum descent algorithm

Let us consider the design of an N-level codebook $C=\{Y_1,Y_2,\ldots,Y_N\}$ for training set $S=\{X_1,X_2,\ldots,X_M\}$, where $Y_i\in R^k$, $i=1,2,\ldots,N$, $X_m\in R^k$, $m=1,2,\ldots, M$, $M>>N$. The maximum descent algorithm views the training vector set S as a global cluster at first. This cluster is split into two new clusters by an optimal partitioning hyperplane. One of these two clusters is then further partitioned into two clusters to generate three new clusters based on a maximum distortion reduction criterion. For the general case of $L(L>2)$ clusters, the MD method attempts to generate $L+1$ clusters by splitting one of the former L clusters into two new clusters and keeping the other L-1 clusters unchanged such that the distortion reduction is maximum. This procedure is performed until the desired number of clusters is obtained. At last, the centriods of these clusters are taken as codewords.

Without loss of generality, we consider the case that the original training set S has been partitioned into L clusters, i.e., $S=\{S_1, S_2,\ldots,S_L\}$. Assume that a hyperplane $H_i(U,v)=\{Z\in R^k\colon U^TZ = v\}$ further partitions the cluster S_i into two non-empty clusters given by

$$S_{ia} = \{Z \in S_i : U^T Z < v\}\,,\quad S_{ib} = \{Z \in S_i : U^T Z \ge v\} \tag{6}$$

where $U\in R^k$, $v\in R$, $1 \le i \le L$, and T denotes the transpose. The centroid of cluster S_i can be defined as $Y_i = \frac{\sum_{j:X_j \in S_i} X_j}{n_i}$ (7)

where n_i is the number of input vectors that belong to cluster S_i. When the centriod Y_i is used to quantize all vectors inside S_i, the induced total squared Euclidean distortion $D(S_i)$ is $D(S_i) = \sum_{j:X_j \in S_i} \left\| X_j - Y_i \right\|^2$ (8)

where $\left\| X_j - Y_i \right\|^2$ denotes the squared Euclidean distortion between X_j and Y_i. Then the reduction in distortion induced by partitioning the cluster S_i into two clusters with the partition plane H_i can be expressed as

$$r_i = D(S_i) - [D(S_{ia}) + D(S_{ib})] \tag{9}$$

Direct computation of the Eq. (9) requires many calculations of squared distances. In order to reduce the computational complexity of Eq. (9), we can easily prove that the Eq. (9) can be rewritten as follows (the proof can be found in reference [4])

$$r_i = n_i \cdot \frac{n_{ia}}{n_{ib}} \cdot \left\| Y_i - Y_{ia} \right\|^2 \tag{10}$$

or

$$r_i = n_i \cdot \frac{n_{ib}}{n_{ia}} \cdot \left\| Y_i - Y_{ib} \right\|^2 \tag{11}$$

where n_i, n_{ia} and n_{ib} are the number of vectors in S_i, S_{ia} and S_{ib}, respectively. Y_i, Y_{ia} and Y_{ib} are centroids of clusters S_i, S_{ia} and S_{ib}, respectively.

From equation (10) we can easily prove that r_i is always greater than zero, i.e., the reduction in distortion is always not less than zero. In order to obtain the maximum reduction function $\hat{r}_i$ for cluster S_i, we need to use some techniques to search the optimal partitioning hyperplane $\hat{H}_i$ for cluster S_i. To form L+1 clusters from the L clusters with maximum distortion reduction, the maximum distortion reduction functions of all clusters need to be calculated (Fig. 1a) and then one of the L clusters S_p which satisfies $\max_{1 \le i \le L}\{\hat{r}_i\} = \hat{r}_p$ is split into two new clusters while the remainder L-1 clusters remain unchanged. In the step of obtaining L+2 clusters from the L+1 clusters, only two maximum distortion reduction functions, i.e., $\hat{r}_{pa}$ and $\hat{r}_{pb}$ of the newly formed clusters S_{pa} and S_{pb} (Fig. 1b) need to be computed since the $\hat{r}_i$ s of all other clusters have already been computed in the previous step. Thus, in each step of forming an additional cluster only two optimal partitioning hyperplanes need to be found. Based on this maximum descent criterion, the clusters are split one by one until the required N clusters are obtained. We can easily prove that only $2N$-3 optimal partition searches are required to design N-level vector quantizer, so the number of optimal partition searches is proportional to the size of codebook.

It is noted that the MD algorithm has the advantage of tending to partition clusters that are densely populated and will never generate the cluster that includes only one vector that sometimes happens in the conventional LBG algorithm. However, the exhaustive searching of the optimal partitioning hyperplane of a multidimensional cluster is a difficult and computationally intensive problem, and it is too hard to carry out in practice. In order to reduce the complexity, three methods of searching the optimal partitioning hyperplane are presented in [4] as follows.

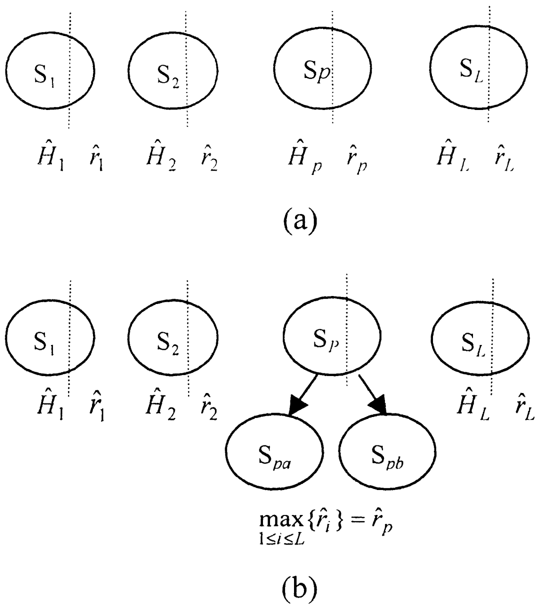

Figure 1. The optimal formation of $(L+1)$ clusters that gives maximum reduction in overall distortion

2.1 Constrained exhaustive search

This method searches the optimal hyperplane among the hyperplanes that are perpendicular to the basis vectors of the vector space that can be obtained by the discrete cosine transform of the input image. For a cluster S_i, the lower and upper bounds of the vectors on one of the DCT basis axes are found. The algorithm starts by assigning a splitting threshold T to the lower bound, and places the training vectors whose projections on this axis are smaller than T in cluster S_{ia} and put the remainder in S_{ib}. The r_i for threshold T is computed. The threshold T is then increased by a predetermined incremental step. The procedures are repeated until T reaches the upper bound. The optimal partitioning hyperplane and the corresponding $\hat{r}_i$ on this axis are recorded. Based on the same procedure aforementioned, all other optimal partitioning hyperplanes for other basis axes are found. By finding the maximum $\hat{r}_i$ among all the $\hat{r}_i$s obtained, the constrained global optimal partitioning hyperplane and the corresponding distortion reduction function are obtained for cluster S_i.

2.2 Successive search

The complexity of the constrained exhaustive search can be reduced by a successive search method based on the idea that the maximum reduction can often be obtained when $D(S_{ia})$ equals to $D(S_{ib})$. This method starts by assigning a threshold T half way between the upper and the lower bounds on an axis. Then $D(S_{ia})$ and $D(S_{ib})$ are evaluated. If $D(S_{ia})>D(S_{ib})$, the threshold is set to T_1 which is half way between T and the lower bound. If $D(S_{ia}) \leq D(S_{ib})$, T_1 is set to a value half way between T and the upper bound. $D(S_{ia})$ and $D(S_{ib})$ are computed for T_1 and the next threshold T_2 can be determined using the same method above. This procedure is repeated until the difference between $D(S_{ia})$ and $D(S_{ib})$ is smaller than a pre-defined value. The optimal hyperplane and the distortion reduction value are then obtained. Similarly, the successive search needs to perform over all the basis axes to find the maximum distortion reduction value and the optimal hyperplane.

2.3 Fast LBG search

If the optimal partition hyperplane is restricted to be perpendicular to only one of the basis vectors of the transformed vector space, we can use the following fast LBG algorithm in the binary splitting of a cluster S_i into S_{ia} and S_{ib} to reduce the amount of computations. In this method, the difference in Euclidean distances $d(X,Y_{ia},Y_{ib})$ between the input vector $X \in S_i$ and the two centroids Y_{ia} and Y_{ib} is defined as follows:

$$d(X,Y_{ia},Y_{ib}) = \sum_{j=1}^{k}(x_j - a_j)^2 - \sum_{j=1}^{k}(x_j - b_j)^2$$

$$= -2\sum_{j=1}^{k}(a_j - b_j)x_j + \sum_{j=1}^{k}(a_j^{\ 2} - b_j^{\ 2}) \qquad (12)$$

where $X=(x_1,x_2,\ldots,x_k)$, $Y_{ia}=(a_1,a_2,\ldots,a_k)$ and $Y_{ib}=(b_1,b_2,\ldots,b_k)$.
Based on the Eq. (12), a vector X is placed in S_{ia} if

$$\sum_{j=1}^{k}(a_j - b_j)x_j > \frac{1}{2}\sum_{j=1}^{k}(a_j^{\ 2} - b_j^{\ 2}) \qquad (13)$$

Otherwise, it is placed in S_{ib}. Since $(a_j - b_j)$ and $\sum_{j=1}^{k}(a_j^{\ 2} - b_j^{\ 2})$ do not change throughout the partitioning process and can be pre-calculated, the amount of computations is greatly reduced.

3 Tabu search based MD algorithm

The basic problem of the MD algorithm is to seek an optimal partition of a certain cluster S into two clusters S_a and S_b, which maximizes the following objective function

$$r = n \cdot \frac{n_a}{n_b} \cdot \|Y - Y_a\|^2 \qquad (14)$$

or

$$r = n \cdot \frac{n_b}{n_a} \cdot \|Y - Y_b\|^2 \qquad (15)$$

Where n, n_a and n_b are the number of vectors in S, S_a and S_b respectively, $n=n_a+n_b$. Y, Y_a and Y_b are centroids of clusters S, S_a, and S_b, respectively.

In order to describe the proposed algorithm, the corresponding indices of the training vectors that belong to cluster S is used to form a solution. Every solution can be divided into two parts. The corresponding training vectors in the first part belong to cluster S_a and the corresponding training vectors in the second part belong to cluster S_b. Thus the optimal partition of S into S_a and S_b can be found using the following algorithm.

Let P_t, p_c and p_b be the test solutions,the best solution of the current iteration and the best solution of all iterations, respectively. Where $P_t=\{p_1,p_2,\ldots,p_{Ns}\}$, $p_i=\{p_{ia},p_{ib}\}$ is one of test solutions, p_{ia} includes the indices of the training vectors belong to S_a and p_{ia} includes the indices of the training vectors belong to S_b, $1 \le i \le N_s$, and N_S is the number of test solutions, $p_c=\{p_{ca},$

$p_{cb}\}$ and $p_b=\{p_{ba}, p_{bb}\}$.

Let R_t, r_c and r_b denote the objective function values for test solutions, the objective function value for the best solution of the current iteration and the objective function value for the best solution of all iterations, respectively. Where $R_t=\{r_1,r_2,\ldots,r_{Ns}\}$, r_i is the objective function of p_i, $1 \le i \le N_s$, N_S is the number of test solutions. The basic idea of the tabu search is to explore the search space of all feasible solutions by a series of moves and to forbid some search directions at the present iteration in order to avoid cycling and jump off local optima. The initail test solutions are generated randomly. After the first iteration, the test solutions are generated from the best solution of the current iteration by some moves. The tabu list memory stores the moved indices only. It is a tabu condition if the moved indicies to generate the test solution from the best solution of the current iteration are the same as any records in the tabu list memory. The algorithm is given as follows:

Step 1. Set the tabu list size T_S, the number of test solutions N_S and the maximum number of iterations I_m. Set the iteration counter i=1 and the insertion point of the tabu list t_i=1. Generate N_S initail solutions $P_t=\{p_1,p_2,\ldots,p_{Ns}\}$ randomly, calculate the corresponding objective values $R_t=\{r_1,r_2,\ldots,r_{Ns}\}$ according to Eq. (14) and find the current best solution $p_c = p_j, j = \arg\max_l r_l$, $1 \le l \le N_s$. Set $p_b = p_c$ and $r_b = r_c$.

Step 2. Copy the current best solution p_c to each test solution p_l, $1 \le l \le N_s$. For each test solution $p_l=\{p_{la}, p_{lb}\}$, $1 \le l \le N_s$, generate a new solution by the following three substeps.

Step 2.1. If $n_{la} \ge 2$ and $n_{lb} \ge 2$, select p_{la} or p_{lb} randomly; else if n_{la} = 1 and $n_{lb} \ge 2$, select p_{lb}; else if n_{lb}=1 and $n_{lb} \ge 2$, select p_{la}; else if n_{la}=1 and n_{lb}=1, neither of them is selected.

Step 2.2. If p_{la} is selected, then select an index in p_{la} randomly and move it to p_{lb}; else if p_{lb} is selected, then select an index in p_{lb} randomly and move it to p_{la}. If neither of them is selected, the solution remains unchanged.

Step 2.3. Calculate the corresponding objective value r_l for the new test solution.

Step 3. Sort r_1, r_2 ,..., r_{Ns} in decreasing order. From the best new test solution to the worst new test solution,if the new test solution is a non-tabu solution or if it is a tabu solution but its objective value is better than the best value of all iterations r_b, then choose this new solution as the current best solution p_c and choose its objective value as the current best objective value r_c, go to step 4; otherwise, try the next new test solution. If all new test solutions are tabu solutions, then go to step 2.

Step 4. If $r_b < r_c$, set $p_b = p_c$ and $r_b = r_c$. Insert the moved index of the current best solution p_c into the tabu memory list. Set the inserting point of the tabu list $t_l = t_l+1$. If $t_l > T_S$, set $t_l = 1$. If $i < I_m$, set $i = i+1$ and go to step 2; otherwise, record the best solution and terminate the algorithm.

4 Performance comparisons and conclusions

In order to demonstrate the efficiency of the proposed algorithm, the LBG algorithm, the conventional MD algorithms and the tabu search based MD algorithm are all implemented to generate codebooks. Here, the parameters of tabu search are T_S=20, N_S=20, I_m=200. All tests are run on Pentium II PC with 233MHz. Two images, LENA and PEPPERS, with resolution 512×512 pixels, 8bits/pixel, are used in this paper. The image LENA is used to generate the codebooks of size 256 with dimension 16(4×4), the image PEPPERS is used to test the coding performance of the codebooks. Results are showed in Table 1, Table 2. From Table 1, we can see that the PSNR of tabu search based MD algorithm is improved by 0.5dB compared with the constrained exhaustive MD algorithm and by 1.4dB compared with the LBG algorithm, although it need more CPU time compared with the constrained exhaustive MD algorithm. For the images outside the training set, the PSNR of tabu search based MD algorithm is improved by 0.2dB compared with the constrained exhaustive MD algorithm and by 0.7dB compared with the LBG algorithm in Table 2. In conclusion, the proposed algorithm can obtain better reconstructed image quality than the LBG algorithm and all conventional MD methods for not only the image in the training set but also the image outside the training set although it needs more CPU time than the LBG algorithm and the conventional MD algorithms. Since the cluster design is operated off line for most of the applications, the proposed tabu search based maximum descent algorithm can be viewed as a good means for cluster generation.

Table 1. Performance comparison for image within training set

Algorithm	CPU time (Sec)	PSNR(dB)
LBG	360.5	30.23
Constrained Exhaustive Search MD	120.2	31.14
Successive Search MD	15.3	30.96
Fast LBG Search MD	3.8	31.08
Tabu Search based MD	1023.1	31.64

Table 2. Performance comparison for image outside training set

Algorithm	PSNR of the image Peppers(dB)
LBG	28.45
Constrained Exhaustive Search MD	28.94
Successive Search MD	28.85
Fast LBG Search MD	28.87
Tabu Search Based MD	29.14

References

[1] Gersho, A. & Gray, R. M. *Vector Quantization and signal compression*, Kluwer Academic Publishers: Boston, 1992.

[2] Linde, Y., Buzo, A. & Gray, R .M. An algorithm for vector quantizer design. *IEEE Trans. on Communications*, **COM-28**, pp. 84-95, 1980.

[3] Ma, C. K. & Chan, C. K. Maximum descent method for image vector quantization. *Electronics Letters*, **27(12)**, pp. 1772-1773, 1991.

[4] Ma, C. K. & Chan, C. K. A fast method of designing better codebooks for image vector quantization. *IEEE Trans. on Communications*, **40(2/3/4)**, pp. 237-242, 1994.

[5] Glover, F. & Laguna, M. *Tabu Search*, Kluwer Academic Publishers: Boston, 1997.