A COMPARATIVE STUDY TO DESIGN A CODE BOOK FOR VECTOR QUANTIZATION

Yoshitaka Takeda, Eiki Noro
Department of Information and Electronic Engineering, Muroran Institute of Technology, Muroran, Japan

Junji Maeda, Yukinori Suzuki
Department of Information and Electronic Engineering, Muroran Institute of Technology, Muroran, Japan

Keywords: Image compression, Vector quantization, Code book optimization, GA, Affinity propagation, Fuzzy.

Abstract: In this paper, we examined six algorithms to construct an optimal code book (CB) for vector quantization (VQ) experimentally. Four algorithms are GLA (generalized Lloyd algorithm), FCM (fuzzy c meams), GA (genetic algorithm), and AP (affinity propagation). The other two algorithms are hybrid methods: AP+GLA and GA+FCM. Performance of the algorithms was evaluated by both *PSNR* (peak-signal-to-noise-ratio) and *NPIQM* (normalized perceptual image quality measure) of decoded images. Computational experiments showed that the performance of each algorithm could be categorized as higher performance and lower performance. GLA, AP and AP+GLA belong to the higher performance group, while FCM, GA and GA+FCM belong to the lower performance group. AP+GLA shows the best performance of algorithms in the higher performance group. Thus, AP+GLA is an optimal algorithm for constructing a CB for VQ.

1 INTRODUCTION

Communication technologies to improve transmission band width are rapidly improving and these technologies have enabled high-speed data transmission. However, the demand for transmission capacity continues to outstrip the transmission band width realized by current technologies. Data compression technologies are therefore being developed for effective use of communication channels. A huge amount of data, including data for characters, voices, music, images and videos is being transferred via communication channels. In these multimedia data, since images and videos need a wide communication band width, effective image and video compression technologies are developing. For still image compression, JPEG (Joint Photographic Experts Group) is used as a de facto standard. It is lossy baseline coding. In JPEG, an image is segmented into 8×8 subimages. The subimages are transformed by DCT (Discrete Fourier Transform). The energy of an image is concentrated into lower frequency components in DCT. This energy compaction provides a good effect to compress images (Gonzalez(2008); Sayood(2000)).

We have been studying vector quantization (VQ) for image compression (Miyamoto(2005); Sasazaki(2008)). In VQ, encoding and decoding an image involves only looking up a code book (CB). Therefore, once the CB is completed, computational cost for image compression is negligible. This is a very attractive point for communication terminals whose computational ability is small. *PSNR* (peak-signal-to-noise-ratio) sharply decreases as compression rate increases in the case of image compression with JPEG. However, in the case of VQ, *PSNR* slowly decreases as compression rate increases (Fujibayashi(2003)). Furthermore, Laha et al. (Laha(2004)) showed that images compressed by VQ provide better *PSNR* than do those compressed by JPEG under the condition of the same bits rate. These are advantageous points of VQ. In principle, since performance of image compression with VQ is determined by a CB, design of a CB is essential for VQ.

A CB is indispensable to carry out VQ, and also quality of a decoded image depends on the CB. In this sense, a CB is essential for VQ. To design a CB, we first prepare learning images. The learning images are segmented into blocks. These blocks constitute learning vectors. To generate CVs, which constitute

Takeda Y., Noro E., Maeda J. and Suzuki Y..
A COMPARATIVE STUDY TO DESIGN A CODE BOOK FOR VECTOR QUANTIZATION.
DOI: 10.5220/0003064200790084
In *Proceedings of the International Conference on Fuzzy Computation and 2nd International Conference on Neural Computation* (ICFC-2010), pages 79-84
ISBN: 978-989-8425-32-4

a CB, the learning vectors are classified into clusters using a clustering algorithm. The prototype of each cluster is a CV. In this paper, we comparatively study clustering algorithm to construct an optimal CB. Algorithms studied in this paper are GLA (generalized Lloyd algorithm), FCM (fuzzy c means), AP (affinity propagation), and GA (genetic algorithm). Two hybrid algorithms, AP+GLA and GA+FCM are also examined. Performance of the algorithms is evaluated by *PSNR* and *NPIQM* (normalized perceptual image quality measure).

The paper is organized as follows. Algorithms to construct a CB are described in section 2. Computational experiments to determine an optimal algorithm for constructing a CB are shown in section 3. Finally, the paper is concluded in section 4.

2 ALGORITHMS TO CONSTRUCT A CB

2.1 GLA and FCM

One the most widely used clustering algorithm is GLA (generalized Lloyd algorithm) (Bezdek(1981)). GLA is a so-called hard clustering algorithm, in which a learning vector is assigned to only one cluster. For clustering, we first determine the number of clusters, k. To formulate the GLA, there are k CVs, $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_k\}$ and M learning vectors, $\mathcal{M} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$. The learning vector $\mathbf{x}_i$ is assigned to the jth cluster when the following equation is satisfied.

$$d(\mathbf{x}_i, \mathbf{y}_j) = \min_{\mathbf{y}_j \in \mathcal{Y}} d(\mathbf{x}_i, \mathbf{y}_j) = \left\|\mathbf{x}_i - \mathbf{y}_j\right\|^2, (i = 1, 2, ..., M). \tag{1}$$

Membership function is degree of belonging to a cluster. Since a learning vector belongs to only one cluster in the GLA, membership function is defined as

$$u_j(\mathbf{x}_i) = \begin{cases} 1(\text{if} d(\mathbf{x}_i, \mathbf{y}_j) = \min\limits_{\mathbf{y}_j \in \mathcal{Y}} d(\mathbf{x}_i, \mathbf{y}_j)) \\ 0(\text{otherwise}) \end{cases}. \tag{2}$$

When clustering is completed, CVs are computed as

$$y_j = \frac{\sum\limits_{i=1}^{M} u_j(x_i) x_i}{\sum\limits_{i=1}^{M} u_j(x_i)} (\forall \ \ j = 1, 2, ..., k). \tag{3}$$

Computation from (1) to (3) is repeated to minimize distortion.

$$J = \sum_{j=1}^{k} \sum_{i=1}^{M} u_j(\mathbf{x}_i) \left\|\mathbf{x}_i - \mathbf{y}_j\right\|^2. \tag{4}$$

The computation ends when J becomes smaller than the predetermined value ε. While the GLA is hard clustering, fuzzy clustering assigns the learning vector to multiple clusters. Formulations are carried out in the same manner of GLA (Bezdek(1981); Baraldi(1999a); Baraldi(1999b); Höppner(1999)).

2.2 Real-coded GA

We designed a CB using a genetic algorithm (GA). GA have been extensively studied to find the global optimal solution in multi-dimensional space for complex problems (Iba(1999)). It is expected that an optimal CB for VQ can be designed using a GA (Hall(1999)). A GA is a stochastic search method and its idea is based on the mechanism of natural selection and genetics. The principal procedures of a GA consists of selection, crossover, and mutation. In a GA, genes are usually coded by binary values: zero or one. This is a bit-string GA. It shows good performance in searching for a solution in a global area. However, the solution is necessarily precise. For this reason, a bit-string GA is used in combination with a local search method. Furthermore, phase structure of a genotype space is much different from that of a phenotype space in a bit-string GA. We select two individuals from parents that are close each other in a phenotype space, and we carry out crossover to produce their children. The children are not necessarily produced in the neighborhood of their parents. Even though a GA finds a promising search area by selection, the crossover may drag the GA away from the area. Thus, a GA does not work well at the middle and last search stages (Kita(1998); Ono(1999)).

To overcome this disadvantage, we employed a real-coded GA in which genes are coded by real values instead of binary values. In the real-coded GA, variable space is continuous, while it is not continuous for the binary-coded GA. This continuity of variable space may produce good results (Kita(1998); Ono(1999)). We also employed the minimal generation gap (MGG) algorithm for selection and simulated binary crossover (SBX) is used to generate a new population (Sato(1997); Deb(1999)). The MGG algorithm could avoid evolutionary stagnation in the last stage of the search, which may be involved by simple GA.

In SBX, we first randomly select two individuals, P_1 and P_2, from N individuals (parents) such as

$$\begin{array}{l} P_1 : x_1^1 x_2^1 \cdots x_k^1 x_{k+1}^1 \cdots x_n^1 \\ P_2 : x_1^2 x_2^2 \cdots x_k^2 x_{k+1}^2 \cdots x_n^2. \end{array}$$

A crossover point is also set randomly. We suppose that genes at the crossover point are x_k^1 and x_k^2. Mean and variance of alleles of these genes are computed as μ_k and σ_k. Then random numbers with a normal distribution are generated by μ_k and σ_k. New genes whose alleles are $\bar{x}_k^1$ and $\bar{x}_k^2$ at the crossover point are produced by this normal distribution such as

$$P_1 : x_1^1 x_2^1 \cdots \bar{x}_k^1 x_{k+1}^1 \cdots x_n^1$$
$$P_2 : x_1^2 x_2^2 \cdots \bar{x}_k^2 x_{k+1}^2 \cdots x_n^2.$$

Then we carry out crossover to generate two children as

$$P_1 : x_1^2 x_2^2 \cdots \bar{x}_k^2 x_{k+1}^1 \cdots x_n^1$$
$$P_2 : x_1^1 x_2^1 \cdots \bar{x}_k^1 x_{k+1}^2 \cdots x_n^2.$$

These procedures are repeated until we generate N children. In the GA with SBX, the distance between two individuals may be large in the early stage of a search. Therefore, the GA can search for a solution in a global area. On the other hand, since the distance between individuals may be small in the last stage of the search, the GA search can be carried out in a local area. In this manner, the shift from a global area to a local area enables an effective search to be carried out.

2.3 Affinity Propagation

We designed a CB using affinity propagation (AP) (Frey(2007)). Computation was carried out based on the programs provided at the website (http://www.psi.toronto.edu/affinitypropagation/).
AP is an effective clustering algorithm that can not only avoid the initial value problem but also realize fast clustering for a large amount of data. Frey and Dueck proposed AP and demonstrated its good performance for clustering tasks such as clustering images of faces, putative exons to find genes, and the problem of identifying a restricted number of Canadian and American cities (accessibility from large subsets of other cities). Most clustering algorithms proposed so far compute cluster centers from the data points forming respective clusters. It is usually a mean of data points in a cluster. The AP algorithm find data points as the cluster centers for respective clusters. This is an essentially different point from other clustering algorithms. There two kinds of data points: exemplar and just data point. An exemplar corresponds to the cluster center of a previous clustering algorithm. Similarity $s(i,k)$ is defined to indicate how well the data point with index k is appropriate to be the exemplar for data point i. It is formulated for points x_i and x_j as

$$s(i,k) = -\left\| x_i - x_k \right\|^2, \tag{5}$$

where indexes i and k indicate a data point and a potential exemplar.

There are two messages, responsibility $r(i,k)$ and availability $a(i,k)$, that are exchanged between data points in the process of clustering. The responsibility is sent from data point i to candidate exemplar k. It is accumulated evidence of how well data point k is appropriate for data point i as an exemplar. The responsibility is sent to all potential exemplars to find the optimal exemplar for point i. $r(i,k)$ is computed during the message exchange as follows:

$$r(i,k) = s(i,k) - \max_{k' \neq k} \left\{ a(i,k') + s(i,k') \right\}, \tag{6}$$

where initial $a(i,k')$ is set to zero. The other message is the availability which is sent from exemplar k to data point i. It is a message of appropriateness from exemplars to data point i to choose k as the exemplar for i. $a(i,k')$ is computed as

$$a(i,k) = \min \left\{ 0, r(k,k) + \sum_{i' \notin (i,k)} \max(0, r(i',k) \right\}. \tag{7}$$

Self-availability is also computed as

$$a(k,k) = \sum_{i' \neq k} \max \left\{ 0, r(i',k) \right\}. \tag{8}$$

3 COMPUTATIONAL EXPERIMENTS

We constructed four CBs using GLA, FCM, GA, and AP. The image consists of four popular images used in image processing: Mandrill, Milk drop, Parrots, and Peppers. The training images are segmented into 4×4 blocks in size to make training vectors. The number of CVs is 256. For both the GLA and FCM, the number of iterations to update CVs is 100. In the FCM, m is set to 1.2. In the GA, N is 30 and T is 600. We selected five test images (Lenna, Earth, Airplane, Sailboat, and Aerial) and encoded these images using CBs constructed by the GLA, FCM, GA, and AP (Takeda(2010)). Performance of the individual clustering algorithms are examined by quality of decoded test images. Image quality is evaluated by both PSNR and NPIQM. PSNR is computed as

$$PSNR = 10 \log_{10} \left(\frac{PS^2}{MSE} \right) (\mathrm{dB}). \tag{9}$$

NPIQM is introduced by Al-Otum (Al-Otum(2003)). The measure is proposed to evaluate perceptual image quality. There is a five-step image quality scale: 1- unacceptable, 2- poor quality, 3- acceptable, 4- good, and 5- pleasant and excellent quality. We also examined hybrid methods. In one method, initial CVs are generated using affinity propagation and then CVs are computed by GLA. In the other method, initial CVs are generated using GA and then CVs are computed by FCM.

Performance evaluation by *PSNR* for each algorithm is shown in Table 1. The performance of each algorithm is categorized as higher or lower performance. The higher performance group consists of GLA, AP and AP+GLA, and lower performance group consists of FCM, GA and GA+FCM. Table 2 shows performance evaluation by *NPIQM* for each algorithm. The performance of each algorithm is also categorized as higher or lower performance. In the same manner as *PSNR*, GLA, AP and AP+GLA belong to the higher performance group, while FCM, GA and GA+FCM belong to the lower performance group. From the two performance evaluations, GLA, AP, and AP+GLA are able to produce a CB with higher quality. AP+GLA shows the best performance. In AP+GLA, initial CVs are generated by AP and clustering of learning vectors is carried out by GLA using those initial CVs. The higher performance of GLA, AP, and AP+GLA is supported by an average distortion. It is computed as

$$D_{ave} = \frac{1}{M}\sum_{i=1}^{M}\min_{\mathbf{y}_j \in \mathcal{Y}} d(\mathbf{x}_i, \mathbf{y}_j), \tag{10}$$

where $d(\mathbf{x}_i, \mathbf{y}_j) = \|\mathbf{x}_i - \mathbf{y}_j\|^2$. $\mathbf{x_i}$ is a vector to be encoded and y_j is a CV. M is the number of vectors to be encoded. Table 3 shows values of D_{ave}. GLA, AP and AP+GLA show smaller values of D_{ave} than those of FCM, GA and GA+FCM. AP+GLA shows the smallest D_{ave}. Figure 1 shows examples of the decoded image “Lenna ”. Corresponding to the results described above, images decoded by the CBs constructed with GLA, AP and AP+GLA show higher quality than those cecoded by the CBs constructed with FCM, GA and GA+FCM. In conclusion, CBs constructed by GLA, AP, and AP+GLA are superior to those constructed by FCM, GA and GA+FCM. The hybrid method AP+GLA is the best method for construting a CB for VQ.

In the computational experiments, AP is an effective method for designing a CB. AP+GLA shows the best performance. AP is a clustering algorithm and it finds a data point as a cluster center. The other clustering algorithms determine the cluster center as an average of data belonging to the cluster. The AP algorithm recursively sends messages to obtain data points that become cluster centers. As stated above, AP determines data points as cluster centers. These data points considered to be good initial cluster center. In our experiments, GLA showed better performance than that of FCM. Both GLA and FCM have an initial value problem, so that clustering depends on initial values. However, since AP gives good initial values for GLA and FCM, AP+GLA shows the best performance. In the GA, we could not obtain good results. The reason is thought to be smaller the number of individuals, 30, in the experiments. $N = 30$ was determined by the basis of computational cost. The GA requires huge computational cost to find good solutions. This relatively small N may not find good solutions. Further study is needed for confirming this speculation.

4 CONCLUSIONS

We constructed four kinds of CB for VQ. GLA, FCM, AP and GA algorithms were used to construct the CBs. Two hybrid algorithms, AP+GLA and GA+FCM, were also employed to construct CBs. The six algorithms were comparatively studied to find the best algorithm. *PSNR* and *NPIQM* were used to evaluate CBs constructed by those algorithms. Computational experiments show that AP+GLA is the best algorithm for constructing a CB.

Table 1: *PSNRs* of decoded images.

	test images				
algorithms	Lenna	Earth	Airplane	Sailboat	Aerial
GLA	27.45	28.12	25.91	26.53	24.84
FCM	26.36	27.82	24.83	24.89	24.50
GA	26.24	27.47	24.74	24.95	24.47
AP	27.39	28.47	26.16	26.47	25.00
AP+GLA	27.52	28.47	26.31	26.71	25.05
GA+FCM	26.32	27.79	24.82	24.89	24.51

Table 2: *NPIQMs* of decoded images.

	test images				
algorithms	Lenna	Earth	Airplane	Sailboat	Aerial
GLA	4.29	4.31	4.06	4.20	4.15
FCM	4.11	4.18	3.97	3.97	4.02
GA	4.12	4.21	4.01	3.98	4.06
AP	4.28	4.31	4.19	4.19	4.15
AP+GLA	4.30	4.35	4.15	4.22	4.16
GA+FCM	4.10	4.18	3.93	3.94	4.02

ACKNOWLEDGEMENTS

This research project is partially supported by grant-in-aid for scientific research of Japan Society for the Promotion of Science (21500211).

Table 3: Average distortion.

algorithms	GLA	FCM	GA	AP	AP+GLA	GA + FCM
average distortiuon	33.25	37.31	38.24	33.28	32.64	37.10

Figure 1: Decoded images of Lenna. The top left image is decoded by GLA. The top middle image is decoded by FCM. The top right image is decoded by GA. The bottom left image is decoded by AP. The top middle image is decoded by AP+GLA. The bottom right image is decoded by GA+FCM.

REFERENCES

R. C. Gonzalez, and R. E. Woods, Digital image processing, Person Prentice Hall/New Jersy, 2008.

K. Sayood, Introduction to data compression, *Morgan Kaufmann Publisher*: Boston, 2000.

T. Miyamoto, Y. Suzuki, S. Saga, and J. Maeda, Vector quantization of images using fractal dimensions, 2005 *IEEE Mid-Summer Workshop on Soft Computing in Industrial Application*, pp. 214-217, 2005.

K. Sasazaki, S. Saga, J. Maeda, and Y. Suzuki, Vector quantization of images with variable block size, *Applied Soft Computing*, vol. 8, pp. 634-645, 2008.

M. Fujibayashi, T. Nozawa, T. Nakayama, K. Mochizuki, M. Konda, K. Kotani, S. Sugawara, and T. Ohmi, A still-image encoder based on adaptive resolution vector quantization featuring needless calculation elimination architecture, *IEEE Journal of Solid-State Circuit*, vol. 38, no. 5, pp. 726-733, 2003.

A. Laha, N. R. Pal, and B. Chanda, Design of vector quantization for image compression using self-organizing feature map and surface fitting, IEEE Tans Image Processing, vol. 13, no. 10, pp. 1291-1303, 2004.

J.C. Bezdek, Pattern recognition with fuzzy objective function algorithms, Plenum: New York, 1981.

A. Baraldi and P. Blonda, Survey of fuzzy clustering algorithms for pattern recognition-Part I, *IEEE Trans Systems, Man and Cybernetics-Part B*: Cybernetics, vol. 29, no. 6, pp. 778-785, 1999(a).

A. Baraldi and P. Blonda, Survey of fuzzy clustering algorithms for pattern recognition-Part II, IEEE Trans Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 29, no. 6, pp. 786-801, 1999(b).

F. Höppner, F. Klawonn, R. Kruse, and T. Runkler, Fuzzy Cluster Analysis, John Wiley & Sons, LTD: New York, 1999.

H. Iba, To solve the mystery of genetic algorithms (in Japanese), Ohmusha: Tokyo, 1999.

L. O. Hall, I. B. Ozyurt and J. Bezdek, Clustering with a generally optimized approach, IEEE Trans Evolutionary Computation, vol. 3, no. 2, pp. 103-112, 1999.

H. Kita, I. Ono, and S. Kobayashi, Theoretical analysis of the unimodal distribution crossover for real-coded genetic algorithms, Evolutionary Computation Proceedings, 1998. *IEEE World Congress on Computational Intelligence*, volume, issue, 4-9, pp. 529-534, 1998.

I. Ono, H. Sato, and S. Kobayashi, Real-coded genetic algorithm for function optimization using the unimodal distribution crossover, *Journal of Japanese Society for Artificial Intelligence*, vo. 14, no. 6, pp. 1146-1155, 1999.

H. Sato, I. Ono, and S. Kobayashi, A new generation alternation model of genetic algorithms and its assessment, J. Japanese Society of Artificial Intelligence, vo. 12, no. 5, pp. 1-10, 1997.

K. Deb, H-G., Beyer, Self-adaptive genetic algorithms with simulated binary crossover. Technical Report No. CI-61/99, University Dortmund, 1999.

B. J. Frey and D. Dueck, Clustering by Passing Messages Between Data Points, Science, vol. 315, pp. 972-976, 2007.

Y. Takeda, Code book optimization with affinity propagation for image compression (in Japanese), Master Thesis of Muroran Institute of Technology, 2010.

H. M. Al-Otum, Qualitative and quantitative image quality assessment of vector quantization, JPEG, and JPEG 2000 compressed image, *Journal of Electronic Imaging*, vol. 12. No. 3, pp. 511-521, 2003.