ELSEVIER

Signal Processing: *Image Communication* 14 (1999) 677–681

SIGNAL PROCESSING: IMAGE COMMUNICATION

www.elsevier.nl/locate/image

Binary vector quantizer design using soft centroids

Pasi Fränti[a,*], Timo Kaukoranta[b]

[a] *Department of Computer Science, University of Joensuu, P.O. Box 111, FIN-80101 Joensuu, Finland*
[b] *Turku Centre for Computer Science (TUCS), Department of Computer Science, University of Turku, FIN-20520 Turku, Finland*

Received 29 May 1997

Abstract

Soft centroids method is proposed for binary vector quantizer design. Instead of using binary centroids, the codevectors can take any real value between one and zero during the codebook generation process. The binarization is performed only for the final codebook. The proposed method is successfully applied for three existing codebook generation algorithms: GLA, SA and PNN.

Keywords: Binary images; Vector quantization; Codebook generation; Clustering algorithms; Image coding

1. Introduction

We consider the clustering problem involved in binary vector quantizer design. The aim is to find a set of *codevectors* (*codebook*) for a given *training set* minimizing the average pairwise distance between the training vectors and their representative codevectors. The question of the proper choice for the training set is not issued here but the motivation is merely to select the best-possible codebook for a given training set.

There are several known methods for generating a codebook [3]. The most cited and widely used is *generalized Lloyd algorithm* (GLA) [5]. It tries to improve an existing initial codebook by iterating the following two steps. In the *partition step*, the training set is partitioned according to the existing codebook. The optimal partitioning is obtained by mapping each training vector to the nearest codevector. In the *codebook step* a new codebook is constructed by calculating the *centroids* of the clusters defined in the partition step.

Unfortunately, the GLA is not well applicable for binary data. It is a descent method that gradually improves the initial codebook until a local minimum is reached. The components of a binary vector, however, can only take two values and therefore gradual changes cannot appear in the codebook easily. Instead, the iteration terminates rather quickly and the quality of the final codebook highly depends on the choice of the initial codebook.

Here, we propose a simple modification to the centroid calculation in the GLA. Instead of using the binarized average vector as the cluster centroid (*binary centroid method*), the components of the codevectors are allowed to take any intermediate values between one and zero during the process.

* Corresponding author. Tel.: + 358 13 251 3103; fax: + 358 13 251 3290; e-mail: franti@cs.joensuu.fi.

0923-5965/99/$ - see front matter
PII: S0923-5965(98)00037-X

The hard thresholding of the centroids is performed only for the final solution in order to obtain binary vectors as the result of the algorithm. The proposed *soft centroids method* does not restrict to the GLA only, but it is applicable to any existing clustering method that uses cluster centroids as the codevectors.

2. Binary vector quantizer design

Let us consider a set of N training vectors consisting of K binary values. The aim of vector quantizer design is to find M codevectors minimizing the average pairwise distance between the training vectors and their representative codevectors. The distance between two vectors X and Y is defined by their absolute distance:

$$d(X, Y) = \sum_{k=1}^{K} |X_k - Y_k|, \quad (1)$$

where X_k and Y_k stand for the kth component of the vectors. In the case of binary vectors this equals to the *Hamming distance*.

Let $C = \{Y^{(j)} | 1 \leqslant j \leqslant M\}$ be a codebook, and $Q(X^{(i)})$ be a mapping which gives the representative codevector in C for a training vector $X^{(i)}$. Then we can define the distortion of the codebook C by

$$\text{distortion}(C) = \frac{1}{N} \sum_{i=1}^{N} d(X^{(i)}, Q(X^{(i)})). \quad (2)$$

A solution for the vector quantizer design can thus be defined by the pair (C, Q). These two depend on each other so that if one of them has been given, the optimal choice of the other one can be uniquely constructed. The GLA applies this property by performing the following two steps in turn starting from an initial codebook.

Partitioning step. The training set is partitioned into M clusters $S^{(j)}$ according to the existing codebook by mapping each training vector to the nearest codevector as defined in Eq. (1):

$$S^{(j)} = \{X^{(i)} | d(X^{(i)}, Y^{(j)}) \leqslant d(X^{(i)}, Y^{(h)}); \quad 1 \leqslant i \leqslant N, 1 \leqslant h \leqslant M\}. \quad (3)$$

Codebook step. A new codebook C' is constructed by calculating the centroids of the clusters defined in the partitioning step:

$$Y'^{(j)}_k = \left[\frac{\sum_{X \in S^{(j)}} X_k}{\sum_{X \in S^{(j)}} 1} \right]_{0/1}, \quad 1 \leqslant j \leqslant M, \quad (4)$$

where $[\bullet]_{0/1}$ denotes binarized value of its parameter. The codevector of a cluster is thus the binarized centroid (average vector) of the training vectors in the cluster.

The problem of the hard thresholding applied in Eq. (4) is illustrated in Fig. 1. Consider a single vector component with a relatively even distribution of zeros and ones among the vectors in a cluster; say 60% zeros and 40% ones. In this case, the corresponding value in the codevector becomes zero. In order to be inverted to the opposite value ($0 \to 1$), more than 10% of the training vectors in the cluster need to be changed, but it does not happen very often in a single partition step of the GLA. The changes generated in the partitioning step are therefore too small to result in remarkable changes in the codebook. Instead, the iteration of the GLA converges rather quickly, which makes the method very sensitive to the initial choice of the codebook. The algorithm converges typically after 3–4 iterations.

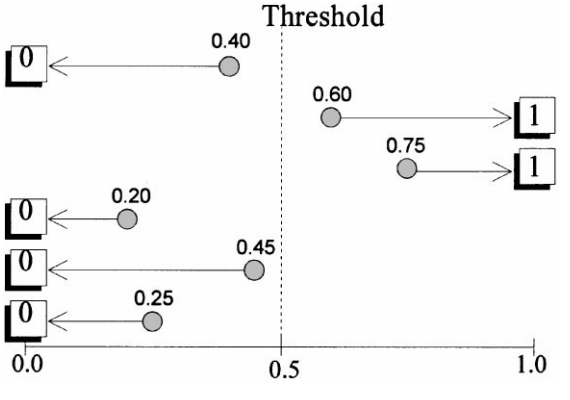

Fig. 1. Hard thresholding of a 6-dimensional average vector (0.40, 0.60, 0.75, 0.20, 0.45, 0.25). The dots represent the average values of the elements and the numbered boxes the thresholded centroid.

3. Soft centroids method

Here, we propose a simple solution to avoid the problem of the hard thresholding. The components of the codevectors are not binarized during the codebook generation, but they are allowed to have any intermediate values between one and zero. The data space is considered as a K-dimensional *Euclidean space*, where each vector component may take any value in the range [0, 1]. Training vectors can appear only in the corner points of the hypercube but the codevectors may be located in any position inside the cube during the process.

For example, consider the example in the previous section. The cluster having 40% ones in the corresponding vector component would have taken the value 0.4 instead of rounding to zero. In this way, gradual changes can happen in the codebook because the changes may take several iterations to climb over the 50% boundary. The hard thresholding, however, is still needed but it is performed only for the final solution in order to obtain binary vectors as the final result of the algorithm.

The proposed *soft centroids method* does not restrict to the GLA only, but it is applicable to any method using cluster centroids as the codevectors. The only modification required is to replace the absolute distance by the (squared) *Euclidean distance* (or *mean-square error*) during the clustering process:

$$d(X, Y) = d_{\mathrm{MSE}}(X, Y) = \sum_{k=1}^{K} (X_k - Y_k)^2 \qquad (5)$$

and to apply the binarization of the codevectors only in the last stage of the algorithm. Another choice would be the use of absolute distances because of binary data. However, this would logically result in the use of median vectors instead of the centroids, which contradicts the idea of the soft centroids. Furthermore, most of the existing methods are designed specifically for the Euclidean distance, which is therefore a natural choice.

4. Test results

Let us consider the following data sets: *Bridge*, *Camera*, *CCITT*-5, *Lates Stappersi*, see Fig. 2. The vectors in the two first sets (*Bridge*, *Camera*) are 4×4 pixel blocks taken from gray-scale images after a BTC-type quantization into two levels according to the mean value of the block [2]. The third data set (*CCITT*-5) is obtained by taking 4×4 spatial pixel blocks from the standard *CCITT*-5 binary test image.

The fourth data set (*Lates Stappersi*) records 270 data samples from pelagic fishes on Lake Tanganyika. The data originates from a research of biology, where the occurrence of 58 different DNA fragments were tested from each fish sample and a binary decision was obtained whether the fragment was present or absent. This data has applications in studies of genetic variations among the species [4]. Here, we consider the clustering of the data as a vector quantization problem.

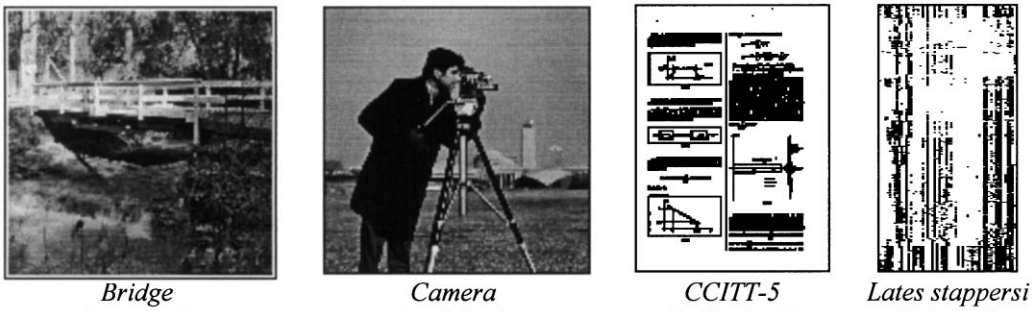

Fig. 2. Sources for the data sets.

The data sets and their properties are summarized in Table 1. In the experiments made here, we will fix the number of codevectors to 256 for the image data sets, and 4 for the DNA data set.

The performance of the GLA is illustrated in Fig. 3 as a function of the number of iterations. The initial codebook is created by selecting M random codevectors from the training set. The binary centroids method converges very quickly, only two iterations are needed in the example. Remarkable improvement is achieved only in the first iteration. Using soft centroids, the codebook develops longer (7 iterations in total), although the main improvement is restricted to the first three rounds. Similar increase of iteration rounds occurs for all data sets. Considering that the binary centroid method can reduce the distortion only about 5% from the random initialization, the further improvement due to the soft centroids method is significant.

The soft centroids method can also be applied in other codebook generation algorithms that uses cluster centroids as the codevectors. The method is

Table 1
Data sets and their statistics

Data set	Vector dimension	# Training vectors	# Codevectors
Bridge	16	4096	256
Camera	16	4096	256
CCITT-5	16	1784	256
Lates Stappersi	58	270	4

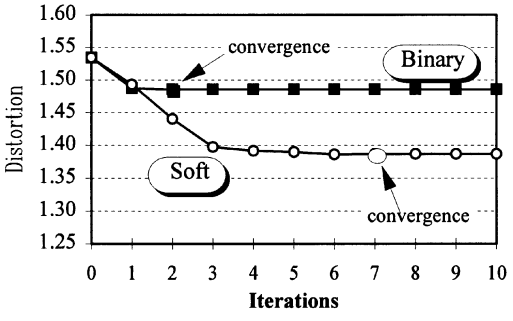

Fig. 3. The distortion of the codebook versus the number of iterations (for *Bridge*).

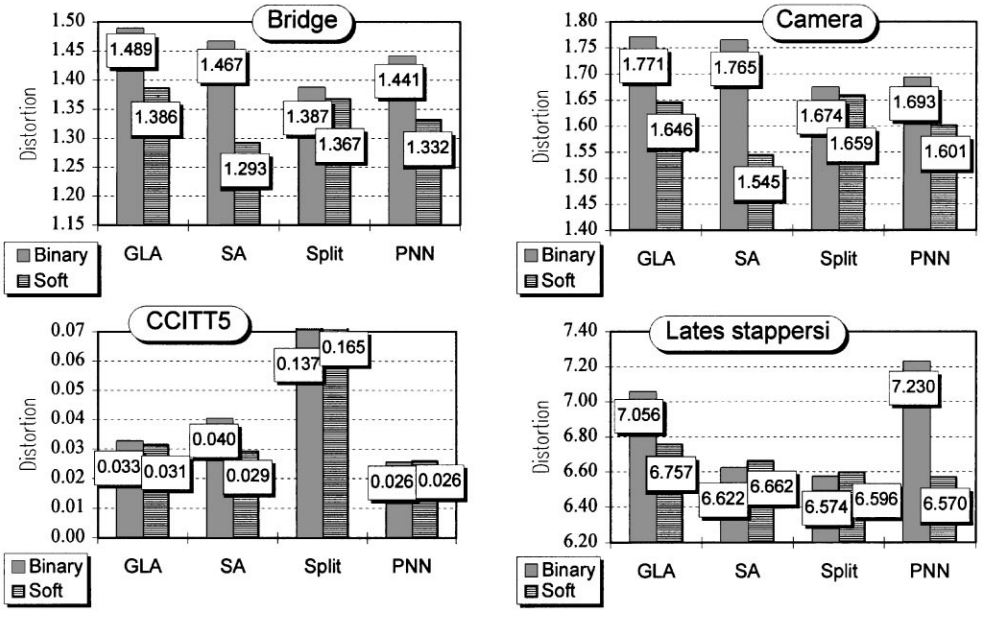

Fig. 4. Performance comparison of the soft centroids method within different codebook generation algorithms. The GLA results are averages from 100 test runs, and the SA results averages from 10 test runs.

studied next within the following algorithms:

GLA = generalized Lloyd algorithm [5]
SA = simulated annealing [7]
PNN = pairwise Nearest Neighbor [1]
Split = iterative splitting method [6]

SA is expected to gain with the soft centroids because it performs same operations to the vectors than the GLA. In the case of the PNN and the splitting method, benefit is expected because the cost of the merge and split operations can be calculated more accurately when soft centroids are applied. Random initialization is used both in the GLA and the SA. PNN is the optimal $O(N^3)$ version (without Kd-tree) of the two variants presented in [1]. The results are summarised in Fig. 4 for the four training sets.

The soft centroids method gives significant decrease in the distortion in almost all cases for the GLA, SA and PNN. The improvement remains marginal (or small increase is obtained) only in cases where the original method already performs rather well. The splitting method for CCITT-5 is the only case where the soft centroids method is not able to help but the result remains unacceptable; it seems that the splitting method is unsuitable for this kind of data set. Overall, the SA and the PNN with the soft centroids are the best choices from the tested methods.

5. Conclusions

Soft centroids method was proposed for binary vector quantizer design. Instead of using binary centroids, the codevectors can take any real value between one and zero during the process. The binarization is performed only for the final codebook. The proposed method was successfully applied with three existing codebook generation algorithms: the GLA, SA and PNN. From the tested methods, splitting algorithm was the only one on which the soft centroids method did not have any significant impact.

Acknowledgements

The work of Pasi Fränti was supported by a grant from the Academy of Finland.

References

[1] W.H. Equitz, A new vector quantization clustering algorithm, IEEE Trans. Acoust. Speech Signal Process. 37 (10) (October 1989) 1568–1575.
[2] P. Fränti, T. Kaukoranta, O. Nevalainen, On the design of a hierarchical BTC-VQ compression system, Signal Processing: Image Communication 8 (6) (September 1996) 551–562.
[3] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, Dordrecht, 1992.
[4] L. Kuusipalo, Genetic differentiation of endemic nile perch Lates stappersi (Centropomidae, Pisces) populations in Lake Tanganyika suggested by RAPD markers, Manuscript 1997, Hydrobiologia, submitted.
[5] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizer design, IEEE Trans. Commun. 28 (1) (January 1980) 84–95.
[6] X. Wu, K. Zhang, A better tree-structured vector quantizer, in: IEEE Proc. Data Compression Conf., Snowbird, Utah, 1991, pp. 392–401.
[7] K. Zeger, A. Gersho, Stochastic relaxation algorithm for improved vector quantiser design, Electron. Lett. 25 (14) (July 1989) 896–898.