# Influence of Vector Quantization on Isolated Word Recognition.

Vincent FONTAINE, Henri LEICH †
Jean HENNEBERT ††

*† Faculté Polytechnique de Mons, 31 Boulevard Dolez, B-7000 MONS, Belgium*
*†† Ecole Polytechnique Federale de Lausanne, CH-1015 LAUSANNE, Switzerland*

**Abstract.** Vector Quantization can be considered as a data compression technique. In the last few years, vector quantization has been increasingly applied to reduce problem complexity like pattern recognition. In speech recognition, discrete systems are developed to build up real-time systems. This paper presents original results by comparing the K-Means and the Kohonen approaches on the same recognition platform. Influence of some quantization parameters is also investigated. It can be observed through the results presented in this paper that the quantization quality has a significant influence on the recognition rates. Surprisingly, the Kohonen approach leads to better recognition results despite its poor distortion performance.

## 1. Introduction

Vector Quantization is almost considered as a data compression technique by DSP researchers. As a matter of fact, a lot of work has been done in this field to improve the quality of the speech- and image coders. The last few years, vector quantization has been applied more and more to reduce the complexity of problems like pattern recognition. In speech recognition, vector quantization can be used to train discrete HMMs. Discrete systems are welcome in real-time implementations since they are less CPU consuming than continuous systems. Results obtained in data compression are due to improvements in vector quantization techniques. However, they are often ignored by people working in speech recognition. The objective of this paper is to show that Vector Quantization methods and their parameters have an important influence on recognition rates. Therefore, some experiments have been done on vector quantization in the framework of a complete speech recognition system. The same recognition algorithms and the same feature vectors have been used to make the comparisons. In this way, we are sure to measure the influence of vector quantization algorithms through the results we report here.

In this paper, we are interested in two approaches. The first one is the classical K-Means approach and the LBG algorithm that are widely used in discrete applications. The second one is the Kohonen self-organizing map. A combination of the two approaches is also reported in this paper.

The paper is divided into five sections. The first one is an introduction, the second one describes the database and the recognition algorithms used in these experiments. The third and fourth sections present the two approaches studied and the results obtained using them. The fifth section is dedicated to a discussion of the results and to some conclusions.

## 2. Tasks and Database

All the experiments reported in this paper were performed on the same recognition algorithm and the same acoustic vectors to ensure reliable comparisons between the vector quantization algorithms.

For these tests, we used a database in American English containing 53 words pronounced by 226 speakers. 190 speakers were used to train the HMMs, 36 speakers were used as test set. The 26 component feature vectors were composed of 12 LPC cepstrum coefficients, their first derivatives and the first and second derivatives of the log-energy. After pre-emphasis (=0.95) and application of a Hamming window, ten LPC coefficients were used to compute the cepstrum. The acoustic vectors were computed every 10 ms over a 30-ms window.

We systematically used four codebooks to quantize the acoustic vectors

In this work, 42 context independent phoneme HMMs have been used. Each phoneme is represented by a 3-state left-to-right model.

## 3. The K-Means approach

A vector quantizer can be viewed as a mapping from the input parameter space (a k-dimensional Euclidian space) into a finite set of N vectors, elements of the parameter space. The reproduction vectors are often called the code vectors or codewords. The set of code vectors forms the codebook.

The mapping operation from a continuous space into a finite set of N codewords generates a quantization error or distortion measure. The definition of the distortion function is guided by the following rules : it must be simple enough so that it can be evaluated in real time and the distortion value must follow a subjective quantization

European Signal Processing Conference (EUSIPCO 94), Edinburgh, UK. 1994. pp. 115-118

quality. A bad quantization should be represented by a high distortion and vice-versa.

In our case, the cepstrum coefficients are often considered as decorrelated so that we can adopt an Euclidian distance as distortion measure for our quantizers :

$$d(x,y) = (x-y)^t . (x-y)$$

An important class of vector quantizers, called Voronoi quantizers or nearest neighbour quantizers, assign an input vector to the code vector that leads to the lowest distortion, i.e. the nearest neighbour. The Voronoi quantizers are known to be the optimal ones in the sense of minimizing the average distortion D = E[d(x,y)]. As a consequence, the optimal partition of the parameter space into N cells (regions of the space where each vector will be quantized by the same code vector) can be computed if the codebook is fixed. These optimal cells are called the Voronoi cells. Inversely, the optimal codebook can be computed if the Voronoi cells are fixed. The code vectors will correspond to the centroids of the N cells for the considered distortion.

This leads to the iterative improvement codebook algorithm : the Generalized Lloyd algorithm or K-Means algorithm. This algorithm is described in two steps :

1. Given a codebook, find the optimal partition of the input space.

2. Given the resulting partition, find the optimal codebook.

Each iteration will reduce the average distortion and the iterative procedure is stopped when the average distortion reduction is small enough [1].

It should be noted that this iterative procedure will not necessarily converge to the optimum codebook but will converge to a local distortion minimum. The initial codebook is then very important for the quantizer quality.

To design a codebook containing $N=2^b$ code vectors, we start with the optimal codebook composed of one centroid, that is the mass center of the input space since we use an Euclidian distance as the distortion measure. The centroid is then replaced by two new code vectors chosen in the centroid neighbourhood, and then K-Means is applied until the algorithm converges to a local average distortion. The two centroids are further splitted into four code vectors and K-Means is applied again. In such a way, we can design codebooks composed of $N=2^b$ centroids and we expect that the average distortion will be close to the global minimum since it started with an optimal codebook.

A drawback of the K-Means quantization method is that we must compare each test vector with all centroids to determine the nearest neighbour. In real-time applications the nearest neighbour search time can be prohibitive.

In order to reduce this search time, Linde, Buzo and Gray developed a tree-structured quantization algorithm [2, 3]. The use of a balanced binary tree ensures a search time of log N through the codebook. This algorithm is known as the LBG algorithm and is still the most common algorithm used in speech recognition.

One problem with this algorithm is that the quantized space is not optimized at each iteration but the K-Means algorithm is applied only to sub-spaces. As a consequence, this algorithm is very sensitive to initial conditions.

Several other techniques (k-d tree based, triangle inequality, ...) have been developed to reduce search times without degradation of the quantization quality [4, 5]. These techniques are very efficient and provide search times compared to the balanced binary tree used in the LBG algorithm [6].

The recognition rates obtained with both K-Means and LBG methods are presented in table 1. The first line of the table gives recognition rates obtained using the classical LBG algorithm trained on 20 speakers. The codebook was splitted when the ratio between distortions of two successive iterations was less than 1% ( $\delta$=0.01).

The second line of the table was obtained by performing the K-Means algorithm at each iteration. The training was also done on 20 speakers and $\delta$ was set to 0.01. For *K-Means 2,* we used the same method as *K-Means 1* except that $\delta$ was set to 0.001.

We can see from this table that some parameters have significant impact on the results. Therefore we see that the use of the pure K-Means algorithm instead of the LBG algorithm improves the results by 2.6%. If we train the codebook with $\delta$=0.001, we improve the results

| **VQ Method** | **Codebooks 1 (32-32-64-64)** | **Codebooks 2 (32-32-128-128)** | **Codebooks 3 (32-32-256-256)** |
|---|---|---|---|
| **LBG** | 73.1 | 76.7 | 77.9 |
| **K-Means 1** | - | 79.3 | 81.4 |
| **K-Means 2** | - | 80.4 | 82.5 |
| **K-Means 3** | - | - | 82.2 |

**Table 1.** Recognition rates obtained using different vector quantization methods. Codebooks i *(a-b-c-d)* means that we used 4 codebooks : *a* centroids for delta energy; *b* centroids for delta-delta energy; *c* centroids for cepstrum; *d* centroids for delta-cepstrum.

furthermore by 1.1%. Finally, if we quantize the cepstrum and delta cepstrum with 256 centroids instead of 128, we see that recognition rates increase by 2.1%.

## 4. The Kohonen approach

The Kohonen Self Organising Feature Map is a neural network trained by following a non-supervised algorithm. The network is made up of $M^n$ neurons, arranged on a *n* dimensional lattice. The neurons are characterised by a coordinate $i = (i_1, i_2, \ldots, i_n)$ with $1 \le i_1, i_2, \ldots, i_n \le M$ and by a synaptic weight vector $\mu_i = \left(\mu_i^1, \mu_i^2, \ldots, \mu_i^d\right)$, where *d* is the network's input dimension. The output response of each neuron *i* to a *d* dimensional input $\xi = (\xi_1, \xi_2, \ldots, \xi_d)$ is given by $\|\xi(t) - \mu_i(t)\|$. At time $t$ [1], an input $\xi(t)$ is presented to the network. The first phase of the training algorithm is the selection of the winner neuron *w* following the condition

$$\|\xi(t) - \mu_w(t)\| = min_i \|\xi(t) - \mu_i(t)\|$$

Any norm $\| \cdot \|$ can be used. The weights are then updated according to the equation

$$\mu_i(t+1) = \mu_i(t) + \alpha(t)\Lambda(w-i,t)\left(\xi(t) - \mu_i(t)\right)$$

where $\alpha(t)$ is the *adaptation gain* $\left(0 < \alpha(t) < 1\right)$ generally decreasing with time and $\Lambda(w-i,t)$ is a *neighbourhood function*, whose value is 1 for $i = w$ and generally decreasing with time and distance to the winner neuron. At each iteration, the updating equation drags the winner neuron and its neighbours towards the input $\xi$. After the training of the map, the weight vectors form a discrete image of the input space and tend to preserve its probability distribution. These weights can then be used as centroids in order to perform a Vector Quantization. As a by-product of the training, a second piece of information is provided; namely the location of winners on the map in the sense that nearby neurons are excited by nearby inputs [6].

In our implementation, $\alpha(t)$ is decreasing linearly with time and the neighbourhood function is decreasing according to a negative exponential of time and distance to the winner neuron. The number of iterations is chosen according to the number of neurons M. The Euclidean norm $\| \cdot \|$ is also used. The dimension *n* of the map is equal to 1 for delta energy and delta-delta energy features. *n* is equal to 2 for cepstra and delta cepstra features.

The training of the map is a self-organising process which tends to provide an image of the probability distribution of the input space. The approach is then quite different to the K-Means where the training criterion is clearly to decrease monotonously the average distortion. Indeed, the distortion produced by a Kohonen Quantizer is known to be greater than the distortion produced by a K-Means' family algorithm.

Our results showed that despite its poor distortion performances, the Kohonen Quantizer provided better recognition rates than in the K-Means approach (typically 2% more). The size of the codebooks also significantly influenced the results. The centroids provided by Kohonen Maps can be used as initial values for the application of the K-Means algorithm. In terms of distortion and recognition rates, this procedure gave better results than the algorithms taken separately. The recognition rates obtained with the different configurations and methods are presented in table 2. These results are totally comparable with those of the K-Means approach since the same features extraction and HMM recognizer have been used.

A large amount of silence is present in the database. Since the Kohonen algorithm tends to preserve the input space probability distribution, too many centroids are allocated for silence. An algorithm able to suppress silence periods from the sample files has been developed [8]. Silence substraction can be performed with different degrees of reliability. When the amount of samples is reduced by 10%, the experiments have showned that it is possible to raise the recognition rate to 85.82% in comparison with 84.13% (first line, second configuration on table 2).

| VQ Method | Codebooks 2<br>(32-32-121-121) | Codebooks 3<br>(32-32-256-256) |
|---|---|---|
| **Kohonen** | 81.34 | 84.13 |
| **Kohonen +<br>K-Means 1** | 81.57 | 84.56 |
| **Kohonen +<br>K-Means 2** | 82.54 | 84.20 |

*Table 2 : Recognition rates obtained using Kohonen and Kohonen+K-Means quantizer.*

[1] The Kohonen algorithm is expressed in the discrete-time formalism.

## 5. Conclusions

The results presented in this paper show that the vector quantization quality has a significant influence on the recognition rates. Two quantization approaches have been analysed and compared inside the same recognition system.

Surprisingly, despite its poor distortion performance, the Kohonen algorithm produces codebooks which provide recognition rates slightly better than the recognition rates obtained when using K-Means family algorithm. Other tests will be carried out on other databases before concluding to the superiority of the Kohonen clustering algorithm against the K-Means algorithms family.

When the K-Means approach is selected to perform the quantization, the results show that the K-Means procedure combined with a fast search algorithm should be preferred to the LBG algorithm.

## 6. Acknowledgements

The authors would like to thank H. Bourlard, J. Vantieghem and S. Accaino from the industry Lernout & Hauspie Speech Products for their contribution to this work.

This work is supported by the Commissions of the European Community through the ESPRIT Project 6488 HIMARNNET.

## 7. References

[1] Allen Gersho et Robert M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, 1992.

[2] G. Zanellato, Reconnaissance de mots isolés dans un contexte multilocuteur, Faculté Polytechnique de Mons, 1989.

[3] Y. Linde, A. Buzo, A.H. Gray, An algorithm for Vector Quantizer Design, IEEE Trans. on Comm. COM-28, N° 1, pp 84-95, 1980.

[4] V. Ramasubramanian and K. K. Paliwal, An optimized K-d Tree for Fast Vector Quantization of Speech, EUSIPCO-88 pp 875-878, 1988.

[5] De-Yuan Cheng, Allen Gersho, A Fast Codebook Search Algorithm for Nearest-Neighbor Pattern Matching, ICASSP-86, pp 265-268, 1986.

[6] Edited Progress Report, ESPRIT P6488 HIMARNNET project, October 1993

[7] Teuvo Kohonen, The Self-Organizing Map, Proceedings of the IEEE, 78(9): 1464-1480, 1990.

[8] Lawrence RABINER and M. R. SAMBUR, "An Algorithm for Determining the Endpoints of Isolated Utterances", The Bell Technical Journal, Vol. 54, no. 2, February 1975, USA.

European Signal Processing Conference (EUSIPCO 94), Edinburgh, UK. 1994. pp. 115-118