Pergamon

Computers Math. Applic. Vol. 30, No. 11, pp. 91–106, 1995
Copyright©1995 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0898-1221/95 $9.50 + 0.00

0898-1221(95)00167-0

Linear Codes Interpolation from Noisy Patterns by Means of a Vector Quantization Process

S. RAMPONE
Dpt. of Fisica Teorica e S.M.S.A., Universitá di Salerno
I-84081 Baronissi (Sa) Italy

(Received January 1995; accepted March 1995)

Abstract—An algorithm inferring a boolean linear code from noisy patterns received by a noisy channel, under the assumption of uniform occurrence distribution over the codewords, and an upper bound to the amount of data are presented. A vector quantizer is designed from the noisy patterns, choosing the obtained codebook as code approximation. It is shown both theoretically and experimentally that, when the data are affected by independent random errors, this strategy requires a small number of patterns to obtain a good identification with high probability of the code from the noisy data.

Keywords—Linear code, Noisy data, Vector quantization, Majority-vote, Probably approximately correct identification.

1. INTRODUCTION

Communication theory deals primarily with systems for transmitting information from one point to another. In information transmission over channels subject to noise disturbances, for example a telephone line, a high frequency radio link, channel noise may corrupt the transmitted signal. The problem is usually faced by encoding the message selected at the source in a redundant way. This allows the decoder, that represents the processing of the channel output, to control the received information. The decoder processing makes use of *a priori* information about the coding [1–4].

In this work, we assume to receive the output of a noisy channel before the decoder, and, for some reason, we do not know the code used by the sender. Our aim is to infer the code only by means of the noisy patterns. As it is defined, the problem is finding a set of reproduction vectors such that a given criterion for the total distortion is minimized, i.e., it is a clustering optimization, or, equivalently, a vector quantizer design problem [5–8].

The questions we address are:

QUESTION 1. Is it possible to minimize the difference between the original code and the inferred one?

QUESTION 2. How many noisy patterns are required?

This work was supported in part by C.N.R., Progetto Finalizzato "Sistemi Informatici e Calcolo Parallelo" under Grant 93.01591 PF 69, by MURST 40% unità I.N.F.M. Università di Salerno, by Contratto Quinquennale C.N.R. - I.I.A.S.S.

The author wishes to dedicate this paper to the memory of E.R. Caianiello, his mentor and inspiring leader, who passed away while this paper was being completed. Furthermore, the author is particularly grateful to M. Marinaro and U. Vaccaro for stimulating discussions, D. BenZvi and A. Esposito for their comments on an earlier draft of this work, G. Papini and A. Feoli for valuable suggestions, and C. D'Apolito, system manager of the Interdepartmental Computing Center of Universitá di Salerno, for her patience.

Typeset by AMS-TEX

The basis of this work is information theory; references [1] and [2] include most of the results we use.

Our approach is strictly connected to vector quantization ([9] is an excellent collection of papers on the matter) and in general to cluster analysis [10]. Although application of vector quantization to the space of binary sequences of fixed length under Hamming distance has been suggested since 1967 [7], and several papers have considered applications of vector quantization to estimation problems in classification, regression, and density estimation, our results appear to be novel.

The problem is also related to "learning from noisy examples," afforded from a theoretical point of view by Angluin and Laird [11] in the case of noise affecting a single bit.

This paper can be summarized as follows. In Section 2, we explain the communication scenario, and the decoding as quantization. In Section 3, we introduce the estimations of the code parameters and of the number of noisy patterns to run an identification procedure, and summarize in an algorithm their computation. In Section 4, we describe a way to find an initial codebook and a refinement algorithm. Finally in Section 5, we show the simulation results obtained with the procedure.

2. DECODING AND QUANTIZATION

We restrict our attention to *Binary Linear codes* [1–4], that are often used in channel encoding, because they are easy to specify, and allow an easy encoding.

An (N, k) binary linear code C is a k-dimensional subspace of the N-dimensional vector space

$$V_N = \{(w_1, w_2, \dots, w_N) \mid w_j \in \{0,1\}\}. \tag{1}$$

Each vector of V_N belonging to C is denoted as

$$\mathbf{w}_h^c = (w_{h,1}^c, w_{h,2}^c, \dots, w_{h,N}^c), \tag{2}$$

and is called *codeword*; N is also called the *length* of the codeword. The codewords, in number of L, are denoted as the vectors

$$\mathbf{w}_1^c, \mathbf{w}_2^c, \dots, \mathbf{w}_L^c. \tag{3}$$

We suppose the codewords transmitted on a *Binary Symmetrical Channel* (BSC) [1,2]. This channel works on binary input and output sequences, where each digit of the input sequence is correctly reproduced at the channel output with some fixed probability $(1 - \varepsilon)$ and is altered by noise into the opposite digit with probability ε, where $\varepsilon < 1/2$. When a codeword $\mathbf{w}_h^c$ is transmitted over this channel, the receiver gets a corrupted version (*noisy pattern*) of the transmitted codeword,

$$\mathbf{w}_i = \mathbf{w}_h^c + \mathbf{z}, \tag{4}$$

where $\mathbf{z}$ is the *error pattern*.

To detect and recover error patterns with minimum mean error probability [2], the *Hamming distance*,

$$d(\mathbf{w}_r^c, \mathbf{w}_s^c) = \sum_{j=1}^{N} |w_{r,j}^c - w_{s,j}^c|, \tag{5}$$

between each pair of codewords $\mathbf{w}_r^c, \mathbf{w}_s^c$, i.e., the number of discordant components, is set to be greater than or equal to an integer quantity $2E + 1$, $E \geq N\varepsilon$. This allows us to define a disjoint *Hamming sphere* of radius E around each codeword [1]. For each codeword $\mathbf{w}_h^c$, we call its sphere *cell* C_h.

Given $\mathbf{w}_i$, if each codeword is sent with the same probability $(1/L)$, the receiver's best strategy for guessing which codeword was sent is to perform the *Maximum Likelihood Decoding* (MLD), mapping $\mathbf{w}_i$ onto the codeword $\mathbf{w}_h^c$ such that the Hamming distance between $\mathbf{w}_i, \mathbf{w}_h^c$ is smallest[1] [1,2].

We say that $\mathbf{w}_i$ is *quantized* as the *reproduction vector* $\mathbf{w}_h^c$ when $\mathbf{w}_i$ belongs to the *Voronoi*, or *nearest neighbor*, region of $\mathbf{w}_h^c$, consisting of all the patterns of the N-dimensional binary space that are closer to $\mathbf{w}_h^c$ than to any other codeword.

In this way, we can see the decoder as an *L-level Vector Quantizer* (L-VQ) [5-7]. An L-level vector quantizer can be defined as a mapping from a source alphabet of N-dimensional vectors to a reproduction alphabet (*codebook*) of L reproduction vectors.

Let us consider this quantizer by a *distortion measure* [8]. A distortion measure is an assignment of a cost of reproducing any input vector $\mathbf{w}_i$ as a codeword $\mathbf{w}_h^c$. When $\mathbf{w}_i$ is quantized as $\mathbf{w}_h^c$, the distortion measure can be defined as a linear function of the Hamming distance

$$\operatorname{dist}(\mathbf{w}_i, \mathbf{w}_h^c) = \frac{1}{N} d(\mathbf{w}_i, \mathbf{w}_h^c). \tag{6}$$

Given such a distortion measure, we can quantify the performance of the system by the average distortion

$$E[\text{dist}]. \tag{7}$$

A vector quantizer is said to be an *optimal* (minimum distortion) quantizer if the average distortion is minimized over all L-level quantizers [5]. In this sense, the MLD L-VQ having C as codebook is optimal, because it minimizes (7) over all L-level quantizers.

Then, in the described environment, our problem can be formally stated as follows: given a set $G = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_P\}$ of P noisy patterns of length N of an unknown linear code C, received from a BSC channel with error probability ε, design from this set an optimal vector quantizer. We call its codebook C^*, and $\mathbf{w}_i^*$ the L^* inferred reproduction vectors.

The described quantizer misinterprets the received pattern if it does not fall inside the Voronoi region of the transmitted codeword. Then in the following, a noisy pattern in the Voronoi region of a codeword $\mathbf{w}_h^c$ will be called *related to* $\mathbf{w}_h^c$, and treated as a noisy version of $\mathbf{w}_h^c$, even if $\mathbf{w}_q^c$, $q \neq h$, has been transmitted.

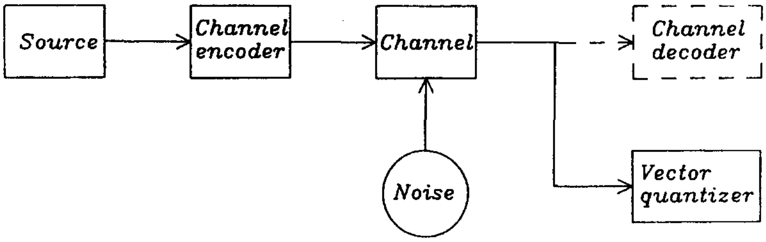


Figure 1. Block diagram of the communication scenario. The messages selected at the source are encoded in a redundant way by the channel encoder and transmitted over the noisy channel. The channel output is received and processed by the quantizer before the decoding processing.

A block diagram of the described environment is reported in Figure 1.

3. PARAMETER ESTIMATION

3.1. Covering Radius

The codebook we have to identify is characterized by cells of Hamming sphere shape. As the first step, we want to estimate the radius E.

[1] Because a noisy pattern can have the same distance from two different codewords, and for computational considerations, the decoding rule is usually restricted to pick $\mathbf{w}_h^c$, when $\mathbf{w}_i$ falls inside a cell C_h, and to detect but not correct the error otherwise. However, because this does not affect the following, we assume the MLD.

The errors of a BSC channel follow the binomial distribution [1], i.e., taking n_e the number of channel errors for a transmitted codeword,

$$\Pr(n_e = k) = (1-\varepsilon)^{N-k}\varepsilon^k \binom{N}{k}, \tag{8}$$

with mean $N\varepsilon$ and variance $N\varepsilon(1-\varepsilon)$. As k goes from 0 to N, the terms (8) first increase monotonically, then decrease monotonically, reaching their greatest value when $k = \lfloor (N+1)\varepsilon \rfloor$ [12]. Thus, if the code C is designed in the hypothesis of the previous section, with respect to the channel noise,

$$E \geq (N+1)\varepsilon, \tag{9}$$

and we can set as lower bound on E

$$E^* = \lceil N\varepsilon \rceil. \tag{10a}$$

To choose an upper bound, it is reasonable to take into account the standard deviation $\sigma = \sqrt{N\varepsilon(1-\varepsilon)}$, having

$$E_u = \lceil N\varepsilon + \sigma \rceil. \tag{10b}$$

Then we choose as estimation of E the mean value

$$E_s = \frac{E^* + E_u}{2}. \tag{11}$$

Figure 2 is a schematic representation of a cell C_h.

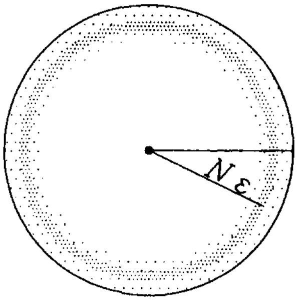


Figure 2. Schematic representation of a cell C_h. The center of the cell is the codeword $\mathbf{w}_h^c$. The channel errors follow the binomial distribution, and the noisy patterns, by having $N\varepsilon$ errors in mean, tend to have a Hamming distance $N\varepsilon$ from the center.

EXAMPLE 1. Suppose given a code C with $N = 32$, $\varepsilon = 0.17$; by (10a) $E^* = 6$, and by (10b) $E_u = 8$; then $E_s = 7$.

3.2. Reproduction Vector

Our aim is to find in the N-dimensional space L regions (*clusters*) C_i^*, and associate with each cluster a reproduction vector $\mathbf{w}_i^*$.

Because the target quantizer is optimal, as necessary condition for the optimality [6], each reproduction vector $\mathbf{w}_i^*$ is chosen to minimize the distortion in cluster C_i^*. This vector is called the generalized *centroid* (or center of gravity or barycenter) of all the patterns lying in C_i^*. Computing the centroid depends on the definition of the distortion measure [5–7]. In the case of Hamming distortion measure (6), that corresponds to the *mean square error*, this centroid is simply the sample mean of the vectors belonging to the cluster C_i^* [6],

$$\frac{1}{m_i}\sum_{r=1}^{m_i} \mathbf{w}_{i_r}, \tag{12}$$

where m_i is the number of patterns $\mathbf{w}_{i_r} \in C_i^*$.

This does not suffice in our case, because the reproduction vectors must belong to $\{0,1\}^N$. To do this we apply a further quantization to the nearest Boolean vector, and in this way the centroid computation,

$$\mathrm{cent}(C_i^*) = \mathbf{w}_i^* = (w_{i,1}^*, w_{i,2}^*, \ldots, w_{i,N}^*), \tag{13}$$

results in the *majority-vote* criterion [1]. By observing that $(1/m_i)\sum_{r=1}^{m_i} w_{i_r,j}$ is greater than $1/2$ if and only if the $w_{i_r,j}$'s in the sum are set to 1 more than 50% of the time, this criterion can be stated as

$$w_{i,j}^* = 1\left[\frac{1}{m_i}\sum_{r=1}^{m_i} w_{i_r,j} - \frac{1}{2}\right], \tag{14}$$

where $1[]$ is the Heaviside function $1[x] = \begin{cases} 1, & \text{if } x > 0; \\ 0, & \text{if } x \leq 0; \end{cases}$.

As we formally see in the following, in a sufficient large set of patterns, each component is unlikely to be in error more than the 50% of the time. In such case, the majority-vote (14) assures the convergence to the transmitted codeword once the patterns related to each codeword have been grouped in a single cluster.

3.3. Sample Size

We have now to evaluate the number of patterns P needed to make use successfully of (14). To this aim we apply the *Probably Approximately Correct* (PAC) criterion [13], recently related to the classical *Estimate of the Error Probability* of pattern recognition literature [14].

The PAC criterion assumes that after randomly sampling *examples* of a *concept* C, an identification procedure should conjecture a concept C^* that with "high probability" is "not too different" from the correct concept. Here, the formal notions of "examples" and "concept" correspond, respectively, to "noisy patterns" and "codebook".

The success of the identification is measured by two given parameters, η and δ, and by the *concept complexity*.

The parameter η, the *tolerance*, is a bound on the "difference" between the conjectured concept C^* and the unknown concept C. We value the difference between C, C^* by,

$$D[C, C^*] = \frac{1}{L^*}\sum_{i=1}^{L^*} D[\mathbf{w}_i^*], \tag{15}$$

where

$$D[\mathbf{w}_i^*] = \frac{1}{N}\min_h d(\mathbf{w}_i^*, \mathbf{w}_h^c) = \min_h \mathrm{dist}(\mathbf{w}_i^*, \mathbf{w}_h^c), \tag{16}$$

i.e., $D[C, C^*]$ is the sample mean of $D[\mathbf{w}_i^*]$.

The parameter δ is a *confidence* parameter that bounds the likelihood that the procedure fails.

The concept complexity is a measure of the number of bits necessary to represent the concept, that, in our case, can be summarized by the length N and by the number of the reproduction vectors L. The parameter L is unknown, but by the *Hamming Bound* [1,2]

$$L \leq \frac{2^N}{\sum_{j=0}^{E}\binom{N}{j}}, \tag{17}$$

setting

$$L_u = \frac{2^N}{\sum_{j=0}^{E_s}\binom{N}{j}}, \tag{18a}$$

we can estimate the maximum reproduction vector number as

$$L_s = 2^{\lfloor \log_2 L_u \rfloor}, \tag{18b}$$

because the number of codewords is an integer power of two. As we see, by (11) and (18), L_s is a function of N and ε.

In this way an identification procedure is said to PAC identify C if and only if the difference (15) between the correct code C and the conjectured codebook C^* is small (less than η) with high probability (greater than $1 - \delta$), given a sample of patterns of size depending on η, δ, N, and ε.

THEOREM 1. *Given a linear code C, $\forall \delta, \eta, \varepsilon \in [0, 1/2[, N \geq 1$, setting*

$$z = \phi^{-1}(1 - \delta), \tag{19}$$

where $\phi()$ is the normal distribution, and

$$\gamma^* = \frac{2L_s\eta + z^2 - z\sqrt{z^2 + 4L_s\eta(1-\eta)}}{2(L_s + z^2)}, \tag{20}$$

$$l^* = -\frac{\ln(\gamma^*)}{(1-2\varepsilon)^2} - \frac{1}{2}, \tag{21}$$

an identification procedure, that collects in clusters the pattern related to each codeword and applies the majority-vote, requires at most

$$P = (2l^* + 1)L_s \ln L_s + 4L_s\sqrt{2l^* + 1} \tag{22}$$

noisy pattern of C, to produce a codebook C^*, such that

$$\Pr(D[C, C^*] < \eta) \geq 1 - \delta. \tag{23}$$

PROOF. First of all we are interested in the sample size P necessary to the acquisition of at least m_i patterns for each codeword. Given the estimation L_s of L (18), we get the expected number of drawings necessary to acquire at least a pattern for each codeword by [12]

$$L_s \ln L_s. \tag{24}$$

Then we evaluate P as

$$P = m_i L_s \ln L_s + 4L_s\sqrt{m_i}. \tag{25}$$

(This claim is proved in the Appendix.)

Now let us consider the patterns related to a codeword collected in a cluster C_i^*. Without loss of generality, we set

$$m_i = 2l + 1. \tag{26}$$

The reproduction vector is selected by the relation (14) that we rewrite as

$$w^*_{i,j} = 1\left[\frac{2}{2l+1}\sum_{r=1}^{2l+1} w_{i_r,j} - 1\right], \tag{27}$$

where $w^*_{i,j}$ is the j^{th} component of the vector. This component is wrong if at least $l + 1$ patterns have an error on the j^{th} component. The probability of this event is

$$p_e = \sum_{h=l+1}^{2l+1} (1-\varepsilon)^{2l+1-h}\varepsilon^h \binom{2l+1}{h}, \tag{28}$$

that we can rewrite as

$$p_e = \sum_{h=\lceil\frac{1}{2}(2l+1)\rceil}^{2l+1} (1-\varepsilon)^{2l+1-h}\varepsilon^h \binom{2l+1}{h}. \tag{29}$$

By the Hoeffding's inequality [15],

$$\sum_{h=\lceil(\varepsilon+s)N\rceil}^{N} (1-\varepsilon)^{N-h}\varepsilon^h \binom{N}{h} \leq e^{-2s^2N}, \qquad \forall\varepsilon, s \in [0,1], \tag{30}$$

setting $s = 1/2 - \varepsilon$, we have,

$$p_e \leq e^{-2(\frac{1}{2}-\varepsilon)^2(2l+1)} = \gamma(l,\varepsilon). \tag{31}$$

The inverse function of $\gamma(l,\varepsilon)$ is

$$l = -\frac{\ln(\gamma)}{(1-2\varepsilon)^2} - \frac{1}{2}. \tag{32}$$

It follows from (31) that the j^{th} component is correct with probability

$$p_c = 1 - p_e \geq 1 - \gamma(l,\varepsilon). \tag{33}$$

Then

$$\Pr(D[\mathbf{w}_i^*] < \beta) \geq \sum_{j=0}^{\lfloor N\beta\rfloor} (1-\gamma(l,\varepsilon))^{N-j}\gamma^j(l,\varepsilon)\binom{N}{j}. \tag{34}$$

The expectation of this distribution is $\gamma(l,\varepsilon)$, see Figure 3, and the variance $\gamma(l,\varepsilon)(1-\gamma(l,\varepsilon))$ [16].

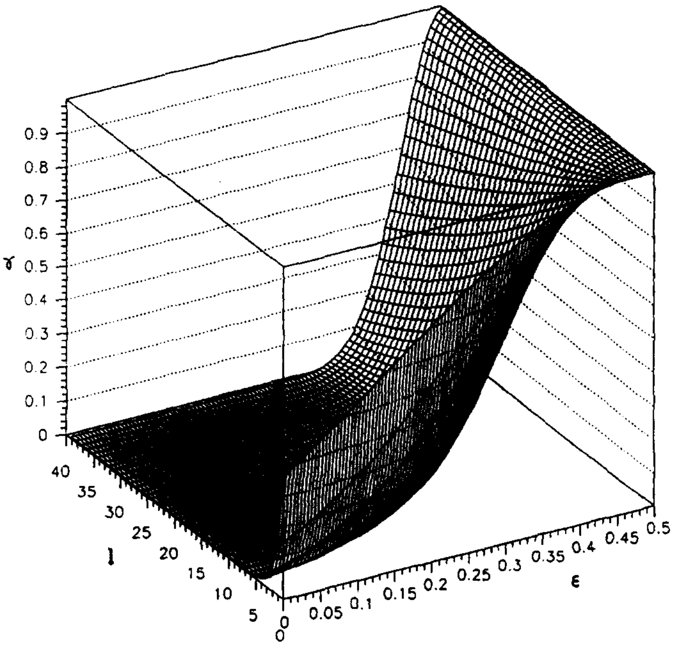


Figure 3. The expected distortion γ varying l and ε.

Because the (15) is the sample mean of $D[\mathbf{w}_i^*]$, by the DeMoivre Laplace theorem [12,16,17]

$$\Pr\left(D[C,C^*] - \gamma(l,\varepsilon) < k\right) \geq \phi\left(k \frac{\sqrt{L^*}}{\sqrt{\gamma(l,\varepsilon)(1-\gamma(l,\varepsilon))}}\right), \tag{35}$$

where ϕ is the normal distribution function

$$\phi(z) = \frac{1}{(2\pi)^{\frac{1}{2}}} \int_{-\infty}^{z} e^{-\frac{1}{2}x^2}\, dx, \tag{36}$$

setting

$$k = \eta - \gamma(l,\varepsilon) \geq 0, \tag{37}$$

we have

$$\Pr\left(D[C,C^*] < \eta\right) \geq \phi\left((\eta - \gamma(l,\varepsilon)) \frac{\sqrt{L^*}}{\sqrt{\gamma(l,\varepsilon)(1-\gamma(l,\varepsilon))}}\right). \tag{38}$$

Calling z the value such that

$$\phi(z) = 1 - \delta, \tag{39}$$

and imposing

$$(\eta - \gamma(l,\varepsilon)) \frac{\sqrt{L^*}}{\sqrt{\gamma(l,\varepsilon)(1-\gamma(l,\varepsilon))}} = z, \tag{40}$$

we have

$$\gamma(l,\varepsilon) = \frac{2L^*\eta + z^2 - z\sqrt{z^2 + 4L^*\eta(1-\eta)}}{2(L^* + z^2)}. \tag{41}$$

The thesis follows substituting to L^* the estimation L_s. (It is easy to prove that the logarithmic loss in (32) of this substitution, when $L^* < L_s$, is balanced by a linear growth of (24).) ∎

3.4. Estimation Algorithm

The evaluations of this section can be summarized in the following procedure.

ESTIMATION ALGORITHM.

```
Step 1: Input: $N$= codeword length, $\varepsilon$ = channel error probability, $\eta$ = tolerance, $\delta$ =confidence.
Step 2: Covering Radius and Cluster number estimation: (Determination of $E_s$ and $L_s$ by (11) and (18))
        2.1 $E^* \leftarrow \lceil N\varepsilon \rceil$
        2.2 $E_u \leftarrow \lceil N\varepsilon + \sqrt{N\varepsilon(1-\varepsilon)} \rceil$
        2.3 $E_s \leftarrow (E^* + E_u)/2$
        2.4 $L_u \leftarrow 2^N / \sum_{j=0}^{E_s} \binom{N}{j}$
        2.5 $L_s \leftarrow 2^{\lfloor \log_2 L_u \rfloor}$
Step 3: Cluster sample size estimation: (Determination of $l^*$ by (19),(20), and (21))
        3.1 determine $z$ such that $\phi(z) = 1 - \delta$
        3.2 $\gamma^* \leftarrow \left(2L_s\eta + z^2 - z\sqrt{z^2 + 4L_s\eta(1-\eta)}\right)/2(L_s + z^2)$
        3.3 $l^* \leftarrow -\ln(\gamma^*)/(1-2\varepsilon)^2 - 1/2$
Step 4: Sample size estimation: $P \leftarrow (2l^* + 1)L_s \ln L_s + 4L_s\sqrt{2l^* + 1}$
Step 5: Output: $E_s$, $L_s$, $P$
```

Table 1. Some values of z for $\phi(z) \in [0.5, 1]$.

$\phi(z)$	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.50	0.00	0.02	0.05	0.07	0.10	0.13	0.16	0.18	0.20	0.23
0.60	0.25	0.28	0.31	0.33	0.36	0.39	0.41	0.44	0.47	0.50
0.70	0.52	0.55	0.58	0.61	0.64	0.67	0.71	0.74	0.78	0.81
0.80	0.84	0.88	0.92	0.95	0.99	1.06	1.08	1.13	1.18	1.23
0.90	1.28	1.34	1.41	1.48	1.55	1.65	1.75	1.88	2.05	2.33

To accomplish the z determination in the Step 3, tables of the values of the standard normal distribution function can be used. Because $\phi(z) = 1 - \delta$ and $\delta \in [0, 1/2[$, the only useful values of $\phi(z)$ are in the range]0.5 ,1]. The Table 1 summarizes some values in this range.

EXAMPLE 2. Suppose given a code C with $N = 32$, $\varepsilon = 0.17$. We want $D[C, C^*]$ lower than 0.1, with probability greater than 0.9. We have:

(1) $N = 32$, $\varepsilon = 0.17$, $\eta = 0.1$, $\delta = 0.1$;

(2) By (11) $E_s = 7$, and by (18) $L_s = 512$;

(3) $\phi(z) = 0.90 \Rightarrow$, from table (1), $z = 1.28 \Rightarrow$, substituting in (20), $\gamma^* = 0.08 \Rightarrow$, substituting in (21), $l^* = 5.18$;

(4) $P = 43177$ (i.e., 1/100000 of the total number of patterns $2^{32} = 4,294,967,296$).

4. CLUSTERING

We have now an estimation of the number of noisy patterns, and a rule to identify, given a cluster, a reproduction vector. We need a way to group the patterns related to each codeword.

The method we adopt to accomplish this task is based on an iterative clustering algorithm known in the pattern recognition literature as the K-means or, in a different version, *LBG* algorithm [6,7,18]. As we need, the algorithm divides the given set of patterns into clusters assigning to each cluster a reproduction vector that minimizes the distortion in that cluster.

4.1. Initial Codebook Design

In its simplest version the algorithm, given an initial set of clusters, assigns each pattern to the cluster having the nearest centroid. Then the centroid is computed again and the process is iterated until no more pattern reassignment take place. The key of the algorithm is the iterative optimization of the initial codebook, and it is well known that the performances essentially depend on this initial choice [6]. The reason is that this method tends to get trapped in local optima and most major changes in assignments tend to occur in the first reallocation step.

In our case, given the pattern distribution (8), we design the initial codebook by a random selection. We take the set G of patterns received from the BSC channel. Iteratively, a pattern (*seed*) is selected within all the pattern of G whose Hamming distance from that pattern is less than $2E^* + 1$. From this set we build a centroid. Then all the chosen patterns are marked and they can not be further eligible as seeds.

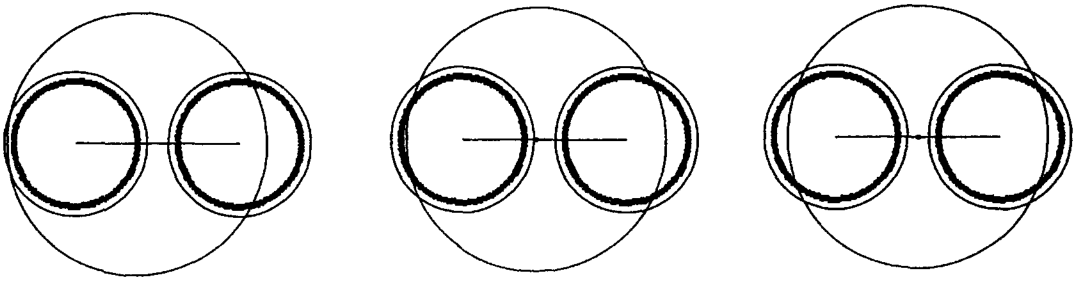
The reason is that if we randomly choose a pattern $\mathbf{w}_i \in G$, we build around this pattern an Hamming sphere C_i^* of radius $2E^*$, and we compute the centroid $\mathbf{w}_i^*$.

(1) *Inside Selection.* By (8) and (9), in the most probable case $\mathbf{w}_i$ is inside a cell C_h (Figure 4a). Then if each C_h contains about the same number of patterns, by the at most complete inclusion of C_h, by (14), $\mathbf{w}_i^*$ approaches $\mathbf{w}_h^c$.

(2) *Outside Selection.* If the pattern is selected outside any sphere,

(a) *Unbalanced Case.* In the most probable case, $\mathbf{w}_i$ is more close to a particular C_h (Figure 4b), and this results in a centroid more close to $\mathbf{w}_h^c$ as in Case 1.

(b) *Balanced case.* Only when there are about the same number of patterns in the major intersections, (Figure 4c), the obtained centroid may go far from any $\mathbf{w}_h^c$. The probability of this event is bounded by the probability that a bounded distance decoder detects the presence of an error pattern but is unable to correct it [19], but, though we are confident it is small, the exact *a priori* evaluation remains an open problem.



(a) Inside selection. (b) Outside unbalanced selection. (c) Outside balanced selection.

Figure 4. The three seed choices in the initial codebook design algorithm.

Anyway, because an outside selection does not cover completely any C_h, another inside pattern can be chosen as seed.

Furthermore, because a linear code is a vector space, $\underline{0} \in C$, and we can select $\underline{0}$ as first seed.

The cluster number, L^*, is determined as part of the clustering procedure. It is initially equal to zero, and is increased by 1 each time a new cluster is selected.

This procedure can be sketched as follows.

INITCODE ALGORITHM.

```
Step 1: Input: $N$= codeword length, $\varepsilon$ = channel error probability, $G = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_P\}$ the set of $P$ received patterns.
        Initialization: Set $L^* = 0$, $C^* = \emptyset$, $E^* = \lceil N\varepsilon \rceil$. Unmark all the patterns in $G$.
Step 2: First seed: $\mathbf{w}_1^* \leftarrow \underline{0}$. Let $C_1^*$ be the subset of $G$ made by all the patterns of $G$ whose distance from $\underline{0}$ is less than $2E^* + 1$. Mark all the patterns in $C_1^*$
Step 3: While $G$ contains unmarked patterns
    3.1. Seed selection: Select an unmarked $\mathbf{w}_i \in G$
    3.2. Cluster selection: Let $C_i^*$ be the subset of $G$ made by $\mathbf{w}_i$ and by all the patterns of $G$ whose distance from $\mathbf{w}_i$ is less than $2E^* + 1$. Let $m_i$ be the $C_i^*$ cardinality. Mark all the patterns in $C_i^*$.
    3.3. Reproduction vector initialization: Build from $C_i^*$ the centroid $\mathbf{w}_i^*$ applying (14)
             $\mathbf{w}_i^* \leftarrow \mathrm{cent}(C_i^*)$.
    3.4. Cluster number updating: Set $L^* \leftarrow L^* + 1$.
    3.5. Codebook updating: $C^* \leftarrow C^* \cup \{\mathbf{w}_i^*\}$.
Step 4: Output: $C^*, L^*$.
```

The resulting C^* is the initial codebook.

4.2. Codebook Refinement

Then we apply a K-means algorithm to refine the codebook.

With respect to the original formulation, at each step the algorithm updates the level number by testing the Hamming distance between the centroids. There are two reasons to do this:

(1) if two or more seed points inadvertently lie near a single cell C_h, their resulting clusters may split C_h;

(2) the existence of an outlier might produce at least one group on the border of the cell.

Moreover, because the reproduction vectors are discrete, and, by (14), reassignments are possible only if at least a centroid changes, the termination test is made on the changes in the codebook.

Finally, the cluster shapes are not required to be Hamming spheres, since these objects do not exhaust the space and a pattern is not guaranteed to go inside any one.

Below, t is the iteration index and $C_i^*(t)$ is the i^{th} shape-free cluster at iteration t, with $\mathbf{w}_i^*(t)$ its centroid. The algorithm is as follows.

Refinement Algorithm.

Step 1: *Input:* $G = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_P\}$ the set of P received patterns; L^* = initial cluster number; C^* = the set of initial reproduction vectors $\mathbf{w}_i^*(0)$, $1 \le i \le L^*$.
Initialization: Set $t = 0$, $E^* = \lceil N\varepsilon \rceil$.

Step 2: *Cluster assignment:* Classify every $\mathbf{w}_i \in G$ into the cluster $C_i^*(t)$ whose centroid $\mathbf{w}_i^*(t)$ is nearest. Let m_i be the cardinality of each resulting $C_i^*(t)$.

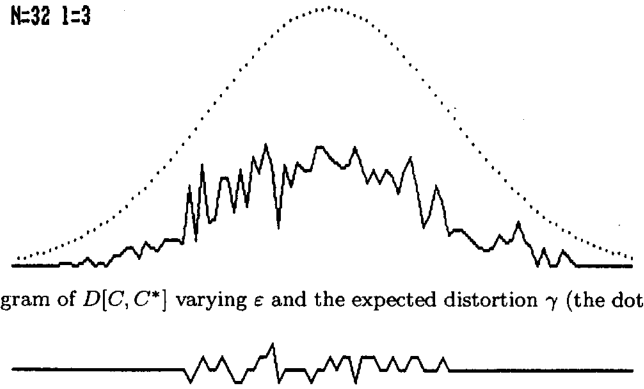
Step 3: *Reproduction vector updating:* $t \leftarrow t + 1$. Update the reproduction vector of every cluster by computing the centroid of the patterns in each cluster as in (14)

$$\mathbf{w}_i^*(t) \leftarrow \operatorname{cent}(C_i^*(t-1)), \quad 1 \le i \le L^*.$$

Step 4: *Level number updating:* If $d(\mathbf{w}_i^*(t), \mathbf{w}_j^*(t)) \le E^*$, $i \ne j$, then erase $\mathbf{w}_j^*(t)$ and decrease the level number L^*.

Step 5: *Termination test:* If the new codebook is the same as the previous, then stop; otherwise go to Step 2.

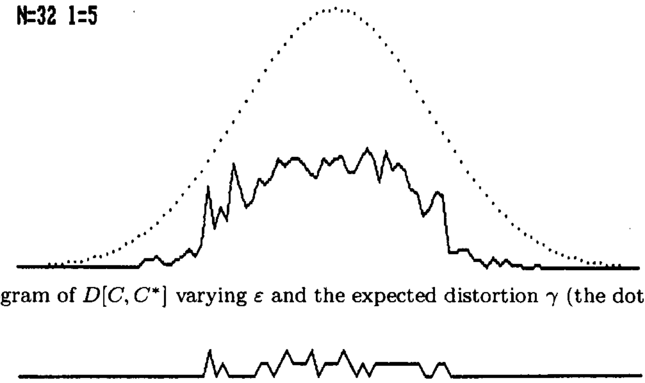
Step 6: *Output:* $\mathbf{w}_1^*(t), \mathbf{w}_2^*(t), \dots, \mathbf{w}_{L^*}^*(t)$.



(a) Histogram of $D[C, C^*]$ varying ε and the expected distortion γ (the dotted line).

(b) Histogram of L^*/L.

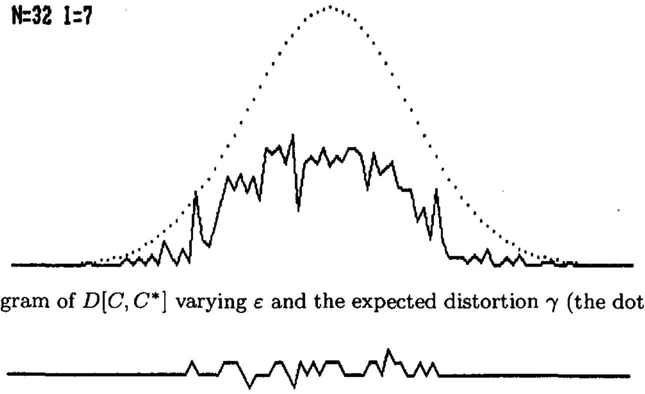
Figure 5. Results of test 1 for l=3.



(a) Histogram of $D[C, C^*]$ varying ε and the expected distortion γ (the dotted line).

(b) Histogram of L^*/L.

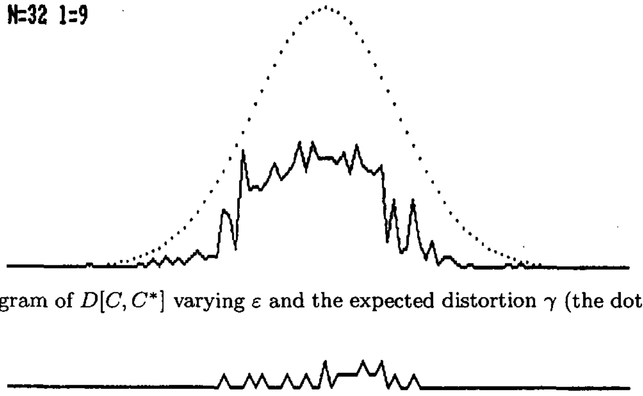
Figure 6. Results of test 1 for l=5.



(a) Histogram of $D[C, C^*]$ varying ε and the expected distortion γ (the dotted line).

(b) Histogram of L^*/L.

Figure 7. Results of test 1 for l=7.



(a) Histogram of $D[C, C^*]$ varying ε and the expected distortion γ (the dotted line).

(b) Histogram of L^*/L.

Figure 8. Results of test 1 for l=9.

5. EXPERIMENTS

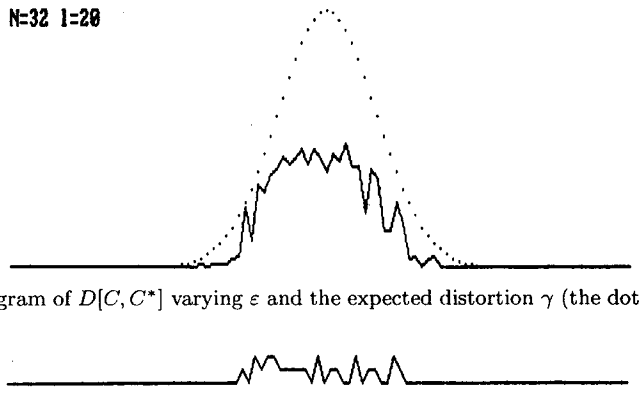
In this section, we compare the behaviour of the algorithm with our derived theoretical predictions. The test environment is the following. Given a code C of L codewords, and a channel error probability ε, a random generator selects a codeword with probability $1/L$. Then each bit of the codeword is modified with probability ε. The process is iterated P times, obtaining the noisy set. Then the InitCode and Refinement algorithms are applied, obtaining the codebook C^*. No *a priori* information is given on the codeword number and on the error correcting capacity.

5.1. Test 1

The aim of the first test is to compare the estimated mean distortion $\gamma(l, \varepsilon)$ (31) and the real distortion measure. To see the results in the largest possible range of ε, we experiment on the situation where we separate two classes of objects, the noisy versions of two complementary codewords of length 32. Fixed P, the codebook is built for all the ε values in $[0, 1]$ with a step of 0.01. In order for the test be meaningful, the first seed assignment is omitted, and E^* is upper bounded to $(N - 1)/2$. For each value, $D[C, C^*]$ is compared with the expected value $\gamma(l, \varepsilon)(31)$. Given parameters:

- Codeword length $N = 32$;
- Channel noise $\varepsilon \in [0, 1]$.

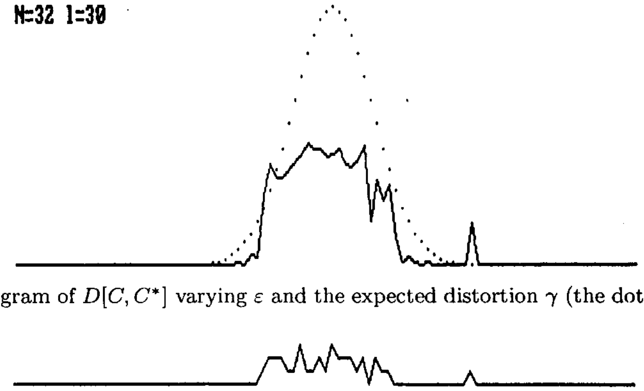
The Figures 5a–10a depict the resulting $D[C, C^*]$'s and the corresponding expected results $\gamma(l, \varepsilon)$ for $l = 3, 5, 7, 9, 20, 30$. The distortion curve is approximately symmetrical because a complementary code is used. The Figures 5b–10b show the ratio L^*/L



(a) Histogram of $D[C, C^*]$ varying ε and the expected distortion γ (the dotted line).

(b) Histogram of L^*/L.

Figure 9. Results of test 1 for l=20.



(a) Histogram of $D[C, C^*]$ varying ε and the expected distortion γ (the dotted line).

(b) Histogram of L^*/L.

Figure 10. Results of test 1 for l=30.

The $\gamma(l, \varepsilon)$ value appears to be an overestimation for $l < 10$ (better than expected), and follows well the distortion curve for $l > 10$.

5.2. Test 2, Reed-Muller (1,3)

In the second serial of tests, we run the algorithm on noisy patterns of the Reed-Muller (1,3) code, of 16 codewords. We want to infer the code with a minimum distortion (less than 0.01, corresponding to an accuracy of 99%) with probability greater than 0.9.
Given Parameters:

- Codeword length $N = 8$;
- Channel noise $\varepsilon = 0.09$;
- Tolerance $\eta = 0.01$;
- Confidence $\delta = 0.1$.

These parameters are given as inputs to the estimation algorithm that estimates the following parameters:

- Error correcting capacity $E_s = 1$;
- Codeword number $L_s = 16$;
- Sample size $P = 1231$.

The random generator then produces the 1231 noisy patterns, and the InitCode and Refinement algorithms are applied, obtaining the inferred code C^*. Figure 11 depicts the results of 10 different runs of the algorithm. The resulting $D[C, C^*]$'s are in each test lower than or equal to the required η, and 7 times the system reaches the perfect identification.

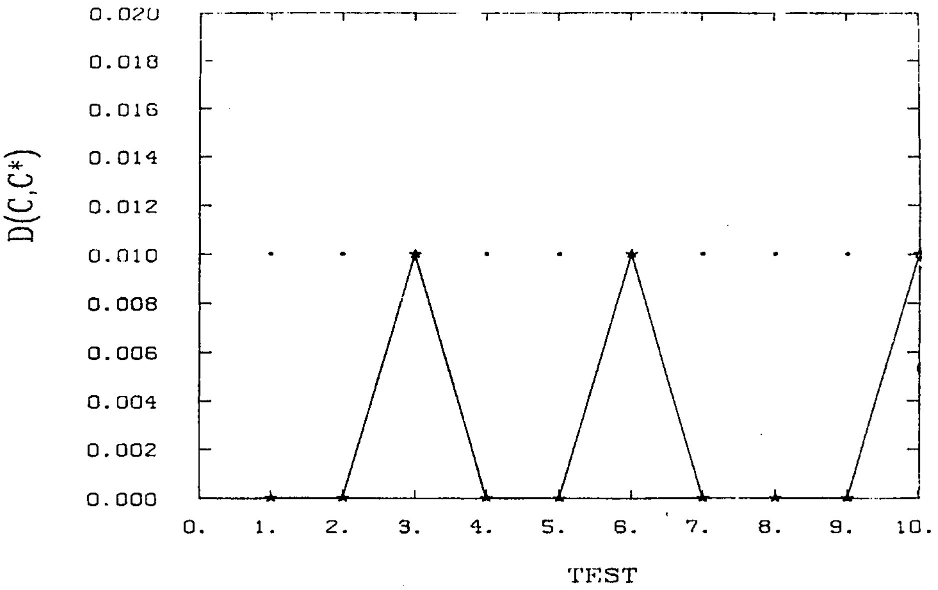


Figure 11. Histogram of $D[C,C^*]$ on ten different runs of the algorithm in test 2. The dotted line is the required bound η.

5.3. Test 3, Reed-Muller (1,5)

The third serial of tests is performed on the Reed-Muller (1,5) code of 64 codewords. In this case, having a high channel noise (about 20%), we require a distortion less than 0.1 with probability greater than 0.9. Given Parameters:

- Codeword length $N = 32$;
- Channel noise $\varepsilon = 0.17$;
- Tolerance $\eta = 0.1$;
- Confidence $\delta = 0.1$.

Estimated Parameters:

- Error correcting capacity $E_s = 7$;
- Codeword number $L_s = 512$;
- Sample size $P = 43177$.

Figure 12 depicts the results of 10 different runs of the algorithm. The resulting $D[C,C^*]$'s are in each test lower than the required η.

6. CONCLUSIONS

The problem afforded in this paper is learning a binary linear code from noisy patterns. As it is defined, the problem is to find a set of reproduction vectors such that a given criterion for the total distortion is minimized, and is thus a clustering optimization, or, equivalently, a vector quantizer design problem.

The main results are an algorithm inferring a binary linear code from noisy patterns and an upper bound to the amount of data.

We derived a general explicit formula that relates the identification accuracy and the sample size, when an extension of the majority-vote criterion is used in the reproduction vector determination. Specifically, the difference between the original and the inferred code decreases exponentially with high probability in the training set size. An application of similar prediction criteria to vector quantization, very different in scope, can be found in [20], but in that case the theoretical worst-case bounds appear to be far from the typically observed performance.

We suggested two heuristic schemes to start and to recover erroneous initializations of a classical clustering procedure. The simulation results show this is an effective approach, and the theoretical results allows us to directly bound the distortion as a function of the codebook training set size.

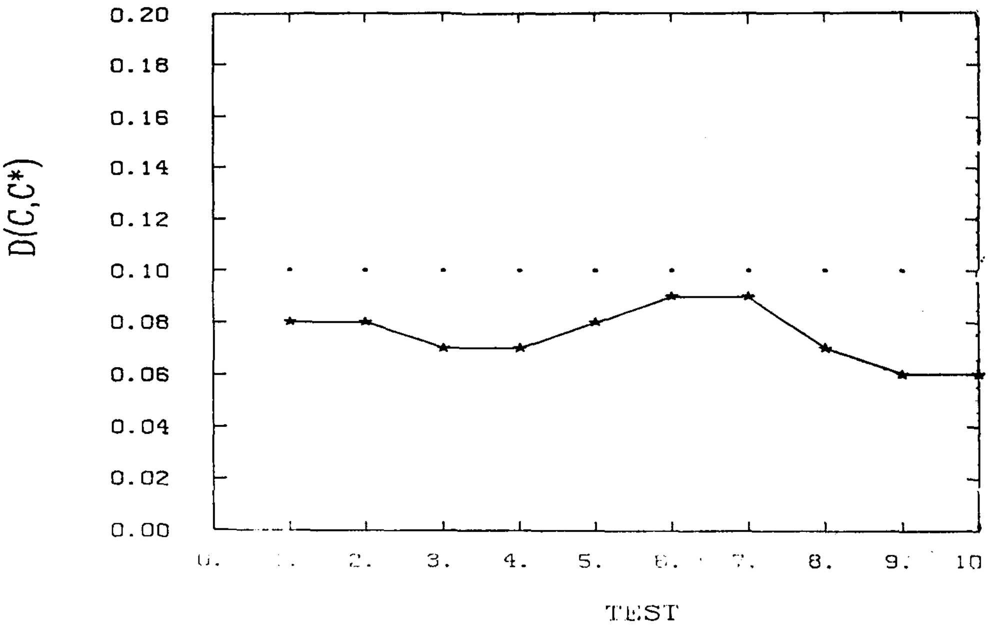


Figure 12. Histogram of $D[C, C^*]$ on ten different runs of the algorithm in test 3. The dotted line is the required bound η.

Open problems rest in the initial codebook design algorithm introduced. An outside balanced selection, in spite of the good results obtained, may in principle cause the identification falls in a local minimum. Many approaches have been suggested to eliminate the sensitivity of K-means to the choice of the initial configuration, for example *simulated* and *deterministic annealing* [21,22], and they will be a matter of a future work.

Moreover, the obtained codebook may not satisfy all the properties of vector subspace, unless the perfect identification is reached. Refinements of low computational cost are necessary. Intuitively, this task can be reached using lattice quantizers [23–25], but they can not be improved by K-means without losing their structure [5].

APPENDIX

In this appendix, we prove the claim (25).

LEMMA 1. *Given a linear code C, with L_s estimated codewords, a sample of*

$$P = mL_s \ln L_s + 4L_s\sqrt{m}$$

acquires at least m patterns in the Voronoi region of each codeword with probability 1.

PROOF. We sample with replacement a population of at most L_s distinct classes uniformly distributed.

Let us consider the sample size necessary for the acquisition of at least one pattern for each class. We call a drawing successful if it results in adding a pattern of a new class in the sample. Let X_i be the number of drawings up to and including the L_s^{th} success. The expected number of drawings necessary to exhaust the entire population is ([12], example IX.3.d)

$$E(X_i) = \mu = L_s \sum_{j=1}^{L_s} \frac{1}{j} \simeq L_s \ln L_s.$$

The variance is

$$var(X_i) = \sigma^2 = L_s \sum_{j=1}^{L_s-1} \frac{j}{(L_s - j)^2} \simeq \frac{4}{5}(2L_s^2 - L_s) - L_s \ln L_s.$$

Now let us consider m repeated samplings for the acquisition of L_s distinct elements

$$X_1, X_2, \dots, X_m.$$

This is a sequence of mutually independent random variables with a common distribution. Let

$$S_m = X_1 + X_2 + \cdots + X_m.$$

By the central limit theorem [12,16,17]

$$\Pr\left(\frac{S_m - m\mu}{\sigma\sqrt{m}} \le k\right) \simeq \phi(k),$$

where ϕ is the normal distribution function. We impose

$$\phi(k) \simeq 1 \quad \Rightarrow \quad k = 3.9.$$

Then

$$\Pr(S_m < m\mu + 3.9\sigma\sqrt{m}) \simeq 1.$$

The thesis follows approximating $3.9\sigma\sqrt{m}$ to $4L_s\sqrt{m}$. ∎

REFERENCES

1. R.J. McEliece, The theory of information and coding, *Encyclopedia of Mathematics and its Applications*, (Edited by G.-C. Rota), Volume 3, Addison-Wesley, London, (1977).
2. R.G. Gallager, *Information Theory and Reliable Communication*, J.Wiley and Sons, New York, (1968).
3. F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error Correcting Codes*, North-Holland, New York, (1981).
4. W.W. Peterson and W.J. Weldon, *Error Correcting Codes*, MIT Press, Cambridge, (1972).
5. R.M. Gray, Vector quantization, *IEEE ASSP Mag.* **1** (April), 4–29 (1984).
6. L. Makhoul, S. Roucos and H. Gish, Vector quantization in speech coding, *Proc. of the IEEE* **73** (11), 1551–1588 (1985).
7. J. MacQueen, Some methods for classification and analysis of multivariate observations, In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, Univ. of California Press, Berkeley, (1967).
8. C.E. Shannon, Coding theorems for a discrete source with a fidelity criterion, *Institute of Radio Engineers, International Convention Record* **7**, 142–163 (1959).
9. A. Huseyin, *Vector Quantization*, IEEE Press, New York, (1990).
10. R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, J. Wiley and Sons, New York, (1973).
11. D. Angluin and P. Laird, Learning from noisy examples, *Machine Learning* **2**, 343–370 (1988).
12. W. Feller, *An Introduction to Probability Theory and its Applications*, Volume 1, Third edition, revised printing, J. Wiley and Sons, New York, (1970).
13. L.G. Valiant, A theory of the learnable, *Comm. of the ACM* **27** (11), 1134–1142 (1984).
14. L. Saitta and F. Bergadano, Pattern recognition and Valiant's learning framework, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **15** (2), 145–155 (1993).
15. W. Hoeffding, Probability inequalities for sums of bounded random variables, *Journal of the American Statistical Association* **58**, 13–30 (1963).
16. A.M. Mood, F.A. Graybill and D.C. Boes, *Introduction to the Theory of Statistics*, McGraw-Hill, Tokyo, (1974).
17. M.A. Golberg, *An Introduction to Probability Theory with Statistical Applications*, Plenum Press, New York, (1984).
18. Y. Linde, A. Buzo and R.M. Gray, An algorithm for vector quantizer design, *IEEE Trans. on Communications* **28** (1), 84–95 (1980).
19. K. Cheung, On the decoder error probability of block codes, *IEEE Trans. on Communications* **40** (5), 857–859 (1992).
20. D. Cohn, E.A. Riskin and R. Ladner, Theory and practice of vector quantizers trained on small training sets, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **16** (1) (1994).
21. A.A. ElGamal, L.A. Emachandra, I. Shperling and V.K. Wei, Using simulated annealing to design good codes, *IEEE Trans. on Inf. Theory* **33**, 116–126 (1987).
22. K. Rose, E. Gurewitz and G.C. Fox, Vector quantization by deterministic annealing, *IEEE Trans. on Inf. Theory* **38** (4), 1249–1257 (1992).
23. A. Gersho, Asymptotically optimal block quantization, *IEEE Trans. on Inf. Theory* **25** (4), 373–380 (1979).
24. J.H. Conway and J.A. Sloane, Fast quantizing and decoding algorithms for lattice quantizers and codes, *IEEE Trans. on Inf. Theory* **28** (2), 227–232 (1982).
25. J.H. Conway and J.A. Sloane, Voronoi regions of lattices, second moments of polytopes and quantization, *IEEE Trans. on Inf. Theory* **28** (2), 211–226 (1982).