Maximum Likelihood Decoding on a Communication Channel

Cesar F. Caiafa†, Nestor R. Barraza‡, Araceli N. Proto†,⋆

† *Laboratorio de Sistemas Complejos, Facultad de Ingeniería, Universidad de Buenos Aires - UBA*
ccaiafa@fi.uba.ar
‡ *Instituto de Ingeniería Biomédica, Facultad de Ingeniería, Universidad de Buenos Aires - UBA*
nbarraza@fi.uba.ar
⋆ *Comisión de Investigaciones Científicas de la Prov. de Buenos Aires Aires - CIC*
aproto@fi.uba.ar

Keywords— **Digital Communication Channel, Maximum Likelihood Decoder, Markov Chain, Logarithmic Distribution, Ising Model.**

Abstract— **A binary additive communication channel with different noise processes is analyzed. Several noise processes are generated according to Bernoulli, Markov, Polya, and Logarithmic distributions. A noise process based on the two dimensional Ising model (Markov 2D) is also studied. In all cases, a maximum likelihood decoding algorithm is derived. We obtain interesting results since in many cases, the most probable code-word is either the closest to the input, or that farthest away, depending on the model parameters.**

I INTRODUCTION

Maximum likelihood - ML decoding on communications has been applied for different kind of channels: Additive White Gaussian Noise - AWGN (Chi-Chao et al (1992), Haykin (2001)), Binary Symmetric Channel - BSC (Haykin (2001)), Binary Erasure Channel - BEC (Khandekar and Mc. Eliece (2001)) and others. ML decoding was also studied when some code is transmitted over the channel, such as Turbo Codes (Hui Jin and McEliece (2002), Moreira and Farrel (2006)), Linear Predicting Code - LPC (Haykin (2001), Moreira and Farrel (2006)) or Cyclic Redundant Codes - CRC (Haykin (2001), Moreira and Farrel (2006)). In some cases, it is considered that Maximum Likelihood is equivalent to minimum Hamming distance decoding. However, it is not true for different kind of noise processes (crossover probabilities). In this paper, we show some cases where the most probably transmitted code-word is that farthest away from the input. Also, cases which are equivalent to minimum Hamming distance, and intermediate possibilities are presented. The type of channel we analyze is the BSC, where the output is produced by adding some noise process to the input code-word. The different kinds of noise distributions we analyze are Bernoulli, Polya contagion, Markov chain and Logarithmic. In addition, a two dimensional Ising noise process is also studied. New and interesting results, depending on the parameters of the noise process, are shown.

II THE BINARY ADDITIVE COMMUNICATION CHANNEL

We study a discrete communication channel with binary additive noise as it is depicted in Fig. 1. Then, the ith output $Y_i \in \{0,1\}$ is the module-two sum of the ith input $X_i \in \{0,1\}$ and the ith noise symbol $Z_i \in \{0,1\}$, i.e. $Y_i = X_i \oplus Z_i$, $i = 1, 2, \cdots$. We assume independence between input and noise processes, and input is a finite code-word chosen from a finite code-book. This type of channel was analyzed in Alajaji and Fuja (1994) where the process Z_i follows the Polya contagion model .

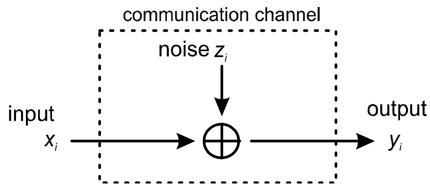

Figure 1: The binary additive communication channel model.

Following these assumptions, for an output vector $\mathbf{Y} = [Y_1, Y_2, \cdots, Y_n]$, a random input code-word $\mathbf{X} = [X_1, X_2, \cdots, X_n]$ and a random noise vector $\mathbf{Z} = [Z_1, Z_2, \cdots, Z_n]$, the channel transition probabilities are given by[1]:

$$P(\mathbf{Y} = \mathbf{y}/\mathbf{X} = \mathbf{x}) = P(\mathbf{Z} = \mathbf{x} \oplus \mathbf{y}) \tag{1}$$

where $\mathbf{x}\oplus\mathbf{y} = [x_1\oplus y_1, x_2\oplus y_2, \cdots]$.

To clarify concepts, a given input, output and noise outcomes could be:

$$\begin{aligned}
\mathbf{x} &= [1,0,0,1,1,0,1,1,1,0,1,0,0] \\
\mathbf{z} &= [0,0,1,1,0,0,0,0,1,1,0,0,1] \\
\mathbf{y} &= [1,0,1,0,1,0,1,1,0,1,1,0,1]
\end{aligned}$$

[1] Through out this paper, we use capital letters for random variable names and lower case letters for denoting realizations of them. Additionally, bold letters are used for vectors.

Therefore, "1's" in the noise process determines which input symbols are changed. The Hamming distance between input and received code-words, is given by:

$$d = \sum_{i=1}^{n} z_i \tag{2}$$

In order to simplify the notation through out the paper, we will avoid the usage of random variable names when a probability of a specific realization is written, for example, instead of writing $P(\mathbf{X} = \mathbf{x}/\mathbf{Y} = \mathbf{y})$ we will write $P(\mathbf{x}/\mathbf{y})$.

III MAXIMUM LIKELIHOOD DECODING

For a code-book $\mathcal{C}$ composed by a set of m code-words, i.e., $\mathcal{C} = \left\{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^m\right\}$, the maximum likelihood decoder chooses, as the estimated input, the most probably code-word $\mathbf{x}^k$ given a received output $\mathbf{y}$, i.e. by maximizing $P(\mathbf{x}^k/\mathbf{y})$. Following the Bayes rule we get:

$$P(\mathbf{x}^k/\mathbf{y}) = \frac{P(\mathbf{y}/\mathbf{x}^k)P(\mathbf{x}^k)}{P(\mathbf{y})} \tag{3}$$

Since $P(\mathbf{y})$ is independent of the decoding rule, and considering that all code-words are equally likely, the ML algorithm results:

$$\hat{\mathbf{x}} = \arg\max(P(\mathbf{y}/\mathbf{x}^k)) : \mathbf{x}^k \in \mathcal{C} \tag{4}$$

Following (1) and (4), the estimated code-word is obtained by choosing $\hat{\mathbf{x}} = \mathbf{x}^k$ which makes $P(\mathbf{z})$ maximum, i.e.:

$$\hat{\mathbf{x}} = \arg\max(P(\mathbf{z}^k)) : \mathbf{z}^k = \mathbf{y} \oplus \mathbf{x}^k,\ \mathbf{x}^k \in \mathcal{C} \tag{5}$$

Then, the estimated input is fully determined by noise (crossover) characteristics and the used code-book.

Following the chain rule of probability, for code-words of length n the noise process can be expressed as:

$$P(\mathbf{z}) = P(z_1) \prod_{i=2}^{n} P(z_i/z_{i-1}, z_{i-2}, ..., z_1) \tag{6}$$

A ML Decoder error probability

If $k\max$ denotes the index for which the probability $P(\mathbf{y}/\mathbf{x}^k)$ is maximized, i.e. $\hat{\mathbf{x}} = \mathbf{x}^{k\max}$, then the conditional error probability of the ML decoder is defined as (Barbero et al (2006)):

$$P(error\ /\mathbf{y}) = P(\mathbf{x}^{k\max} \neq \mathbf{x}^k/\mathbf{y}) \tag{7}$$

and the error probability of the ML decoder is

$$P(error) = \sum_{\mathbf{y}} P(error\ /\mathbf{y})P(\mathbf{y}) \tag{8}$$

Now we obtain an expression of the error probability in terms of the code-book $\mathcal{C}$ and the received vector $\mathbf{y}$. Equation (7) can be rewritten as follows:

$$P(error\ /\mathbf{y}) = \sum_{i \neq k\max} P(\mathbf{x}^i/\mathbf{y})$$

and by using the Bayes rule, it is easy to see that the conditional error probability can be written in the following form:

$$P(error\ /\mathbf{y}) = \frac{\sum_{i \neq k\max} P(\mathbf{y}/\mathbf{x}^i)P(\mathbf{x}^i)}{\sum_{i=1}^{m} P(\mathbf{y}/\mathbf{x}^i)P(\mathbf{x}^i)} \tag{9}$$

$$= \frac{1 - P(\mathbf{y}/\mathbf{x}^{k\max})}{\sum_{i=1}^{m} P(\mathbf{y}/\mathbf{x}^i)P(\mathbf{x}^i)} \tag{10}$$

From equation (10), it is clear that, as the flatter the function $P(\mathbf{y}/\mathbf{x}^k)$ is in terms of $\mathbf{x}^k$, the bigger the error is.

In the following subsections, we analyze the decoder behavior for some specific noise distributions.

B Bernoulli noise model

For this noise distribution, all the Z_i's are independent and have a common parameter p (probability of change in one bit or crossover probability), so (6) results:

$$P(z_i/z_{i-1}, z_{i-2}, ..., z_1) = P(z_i) = p^{z_i}(1-p)^{1-z_i} \tag{11}$$

According to (1), (6) and (11), the probability that a given code-word $\mathbf{x}^k$ had been input when a code-word $\mathbf{y}$ is received $P(\mathbf{y}/\mathbf{x}^k)$ is given by:

$$g_B(d) = P(\mathbf{z}^k) = \binom{n}{d}\left(\frac{p}{1-p}\right)^d (1-p)^n \tag{12}$$

where $d = d(\mathbf{x}^k, \mathbf{y})$ is the Hamming distance between $\mathbf{x}^k$ and $\mathbf{y}$ as was already defined in (2).

As it can be seen from (12), when p is less than $1-p$, the most probable input code-word (ML decoding) which maximizes $g_B(d)$ is that one closest from that received (minimum d). Conversely, when p is greater than $1-p$, the ML input decoding is that having the greatest d, i.e., the most different code-word from that received. This simple case shows the two possibilities for ML decoding, when $p < \frac{1}{2}$, the noise parameter is not enough to produce considerable changes. When $p > \frac{1}{2}$, the noise parameter is big enough to consider that the input was changed at maximum.

C Polya contagion noise model

As it was analyzed in Alajaji and Fuja (1994), when the noise process is given by the Polya contagion model (see Polya and Eggenberger (1923), Feller (1950)) the conditional probabilities are given by:

$$P(z_i/z_{i-1}, z_{i-2}, ..., z_1) = P(z_i/s_{i-1}) \tag{13}$$

where $s_{i-1} = \sum_{l=1}^{i-1} z_i$. The channel transition probabilities result:

$$g_P(d) = P(\mathbf{z}^k) = \frac{\Gamma(1/\delta)\Gamma(\rho/\delta + d)\Gamma(\sigma/\delta + n - d)}{\Gamma(\rho/\delta)\Gamma(\sigma/\delta)\Gamma(1/\delta + n)} \tag{14}$$

where d is the Hamming distance as it was defined before; ρ, σ and δ are model parameters and $\Gamma(t) = \int_0^\infty u^{t-1}\exp(-u)du$ is the gamma function. Since $g_P(d)$ is strictly convex, has a unique minimum d_0, and is symmetric about d_0, the most probable code-word will be either that having minimum or maximum Hamming distance from the received code-word (Alajaji and Fuja (1994)). It means, the best estimate corresponds to d farthest away from d_0. This property for the Polya contagion model, is independent from the parameters, it means, the estimated input could be the closest or the farthest depending on the received code-word. It is due to the convexity of $g_P(d)$.

D Markov noise model

We consider here that the noise process can be modeled by a first order Markov chain (Feller (1950)), i.e. $P(z_i/z_{i-1}, z_{i-2}, ..., z_1) = P(z_i/z_{i-1})$. This model depends on three parameters: the crossover probability $p = P(z_i = 1)$ and the noise transition probabilities $\alpha = P(z_i = 1/z_{i-1} = 0)$ (probability of a bit "1" given that the previous noise outcome is a "0") and $\beta = P(z_i = 0/z_{i-1} = 1)$ (probability of a bit "0" given that the previous noise outcome is a "1"). Using the chain rule (6) we obtain the channel transition probabilities as follows:

$$P(\mathbf{z}^k) = p^{z_1}(1-p)^{1-z_1}\alpha^{n_{10}}(1-\alpha)^{n_{11}}\beta^{n_{01}}(1-\beta)^{n_{00}} \tag{15}$$

where the parameters n_{st} $(s, t = 0, 1)$ are the number of bits with the value "s" followed by a bit with the value "t" and verifying the constraint: $n_{10} + n_{11} + n_{01} + n_{00} = n - 1$.

A very simple expression for the ML decoder is obtained from (15) for the particular case where the noise transitions are symmetric, i.e. $\alpha = \beta$. In the later case, the function to be maximized (ML decoder) is:

$$g_M(z_1, q) = \left(\frac{p}{1-p}\right)^{z_1}\left(\frac{\alpha}{1-\alpha}\right)^{q} \tag{16}$$

where $q = n_{01} + n_{10}$ is the number of transitions ("0" to "1" and "1" to "0") in the noise vector $\mathbf{z} = \mathbf{y} \oplus \mathbf{x}^k$. We conclude from (16) that the ML decoder is a non-decreasing (decreasing) function of q when the noise transition probability is $\alpha > 0.5$ ($\alpha < 0.5$). In other words, when $\alpha > 0.5$, the most probable input code-word is that one corresponding to a noise vector with the highest number of transitions as possible (maximum q).

E Logarithmic noise model

In this model, we consider that noise is composed by alternate chains of "1's" and "0's" and the length of each chain follows a logarithmic distribution (Douglas (1980)). If we denote by U the length of a given "1's" chain and by V the length of a given "0's" chain, then:

$$P(U = u) = \frac{-\beta_1}{u\ln(1-\beta_1)} \tag{17}$$

$$P(V = v) = \frac{-\beta_0}{v\ln(1-\beta_0)} \tag{18}$$

where β_1 and β_0 are the parameters of the logarithmic distributions corresponding to "1's" and "0's" respectively and $0 < \beta_1, \beta_0 < 1$.

In order to clarify this model, a noise output example is shown below:

$$\mathbf{z} = [\underbrace{0,0}_{v_1=2}, \underbrace{1,1}_{u_1=2}, \underbrace{0,0,0,0}_{v_2=4}, \underbrace{1,1}_{u_2=2}, \underbrace{0,0}_{v_3=2}, \underbrace{1}_{u_3=1}]$$

where u_i and v_j are the lengths of the "1's" chain number i and the "0's" chain number j. Notice that high values of β_1 and β_0 produce noise configurations with long chains, on the other hand, for $\beta_1, \beta_0 \to 0$ we get configurations with alternate single "1's" and "0's".

The interest in this model comes from the property that the probability of getting a "1" in a given bit, following a group of r "1's", tends to a constant value β_1 as $r \to \infty$, as it is shown from the conditional probability:

$$P(Z_i = 1/Z_{i-1} = 1, \cdots, Z_{i-r} = 1) = \frac{S_{r+1} + S_{r+2} + \cdots}{S_r + S_{r+1} + \cdots} \tag{19}$$

where:

$$S_r = P(U = r)$$

This property shows a difference with the Polya contagion model, where the conditional probability (19) tends to 1 as r tends to infinity. The property (19) for the logarithmic distribution was remarked in Siromoney (1964).

Assuming independence among u_i and v_j $\forall\, i, j$, we obtain the channel transition probabilities as follows:

$$P(\mathbf{z}^k) = \left(\prod_{i=1}^{k_1} P(u_i)\right)\left(\prod_{j=1}^{k_0} P(v_j)\right) \tag{20}$$

where k_1 is the number of "1's" chains, k_0 is the number of "0's" chains,

In order to obtain the ML decoder, we apply the natural logarithmic function to (20) and we obtain the $g_L(.)$ function to be maximized which is:

$$g_L(n_1, k_1, k_0, \{u_i\}, \{v_j\}) = n_1 \ln\left[\frac{\beta_1}{\beta_0}\right] \tag{21}$$

$$-k_1\ln[\gamma_1] - k_0\ln[\gamma_0] - \sum_{i=1}^{k_1}\ln[u_i] - \sum_{j=1}^{k_0}\ln[v_j]$$

where $n_1 = \sum_{i=1}^{k_1} u_i$ is the total number of "1's" and the parameters γ_1 and γ_0 are defined by $\gamma_s = -\ln(1-\beta_s)$ (for $s = 0, 1$).

From the observation of equation (21) we conclude that there are too many variables to measure ($n_1, k_1, k_0, \{u_i\}$ and $\{v_j\}$) for the implementation of the ML decoder, which could be a problem from the point of view of its complexity. For this reason, in this paper, we propose an approximation of (21) in order to reduce its

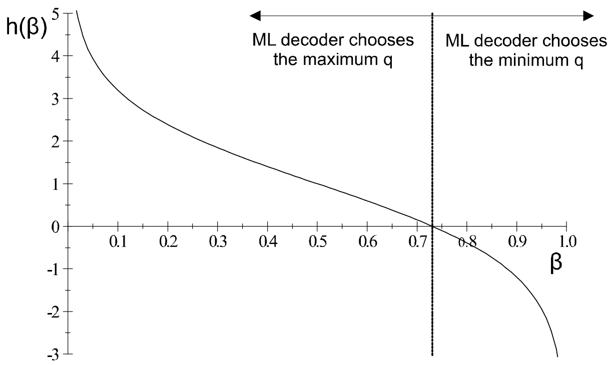

Figure 2: Plot of $h(\beta) = 1 - \ln\left[\frac{\beta}{1-\beta}\right]$. The ML decoder chooses the maximum or minimum number of transitions q as $\beta < 0.73$ or $\beta > 0.73$, .

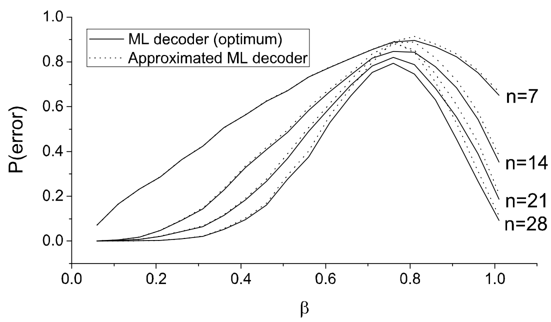

Figure 3: Exact ML decoder (optimum) versus Approximated ML decoder for $\beta = \beta_1 = \beta_0$, $m = 16$ and $n = 7, 14, 21$ and 28.

complexity. The idea is that the last two terms in (21) can be approximated by using the linear approximation of the logarithm ($\ln(t) \approx t - 1$ for $|t| < \epsilon$) as follows:

$$\sum_{i=1}^{k_1} \ln u_i \approx \frac{n_1}{\mu_1} - k_1 + k_1 \ln \mu_1 \quad (22)$$

$$\sum_{i=1}^{k_0} \ln v_i \approx \frac{n - n_1}{\mu_0} - k_0 + k_0 \ln \mu_0 \quad (23)$$

where $\mu_1 = E[U] = -\beta_1 / [(1-\beta_1)\ln(1-\beta_1)]$ and $\mu_0 = E[V] = -\beta_0 / [(1-\beta_0)\ln(1-\beta_0)]$ are the mean values of the logarithmic random variables U and V respectively. Note that, in the approximations (22) and (23), we have used the approximation for $\left|\frac{u_i}{\mu_1}\right|, \left|\frac{v_j}{\mu_0}\right| < \epsilon$ which indicates that these approximations will be valid for the cases where chain lengths are not so far away from their mean values. Finally, by putting (22) and (23) in (21), we obtain the approximated $g_L(.)$ function which is:

$$\hat{g}_L(n_1, k_1, k_0) = n_1 \ln\left[\frac{\beta_1}{\beta_0} + \frac{1}{\mu_1} - \frac{1}{\mu_0}\right] \quad (24)$$
$$+ k_1 \{1 - \ln[\gamma_1 \mu_1]\} + k_0 \{1 - \ln[\gamma_0 \mu_0]\}$$

Let us now consider a simple case where "1's" chains and "0's" chains are identically distributed, i.e. $\beta = \beta_1 = \beta_0$. In this case, $\mu_1 = \mu_0$ and $\gamma_1 = \gamma_0$ and therefore the approximated ML decoder is even simpler:

$$\hat{g}_L(q) = (q+1)\left\{1 - \ln\left[\frac{\beta}{1-\beta}\right]\right\} \quad (25)$$

where $q = k_1 + k_0 - 1$ is the number of transitions ("0" to "1" and "1" to "0") in the noise vector $\mathbf{z} = \mathbf{y} \oplus \mathbf{x}^k$. Looking at equation (25) we see that, in this particular case, $\hat{g}_L(q)$ depends on the number of transitions linearly; so, we need to determine if the factor $h(\beta) = \left\{1 - \ln\left[\frac{\beta}{1-\beta}\right]\right\}$ is positive or negative in order to assign the most probable transmitted code-word to the maximum or to the minimum number of transitions q. From Fig. 2 we see that $\beta = 0.73$ is a threshold from the most probable code-word corresponds to that having maximum or minimum number of transitions q, provided $\beta < 0.73$ or $\beta > 0.73$.

In order to test the effectiveness of our approximations (22) and (23) we have conducted a huge number of simulations where vector noises were generated according to their logarithmic distributions for the case of having $\beta = \beta_1 = \beta_0$ and covering the complete range of the parameter β. A random code-book with $m = 16$ code-words was generated for different cases of code-word lengths n ($n = 7$, 14, 21 and 28) and a minimum Hamming distance among code-words of $d(\mathbf{x}^i, \mathbf{x}^j) = 2$ was guaranteed. For each value of β, a total of 500 simulations were conducted in order to average the obtained decoder error probability and reach to an estimation of (8). In Fig. 3 the decoder error probability obtained by using the exact ML decoder (equation (21)) and the approximated ML decoder (equation (25)) are shown. Notice that the exact ML decoder always gives a lower error probability than the approximated version as expected. Maximum probability error is reached at $\beta \approx 0.75$, as shown. We remark that this value of β also gives the maximum variance of q, in agreement with the maximum probability of error of the ML decoder and the transition threshold shown in Fig. 3. These results will be further studied in a future work.

F 2D Ising noise model

In this subsection, we extend the ML decoder for 2D binary signals transmitted over a channel with the same characteristics as shown in Fig. 1. 2D signals are useful for representing digital images. A very well known model for binary images is the Ising model which has its roots in statistical mechanics as a model for ferromagnetic materials (Huang (1987)). The Ising model has been widely applied to model interactions between pixels in images, (Geman and Geman (1984)), and introduced the development of the theory of Markov Random Fields, (Greaffeath (1976)). In this paper we propose to use the

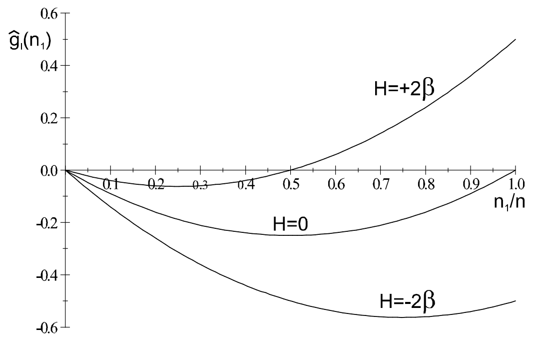

Figure 4: Likelihood function for 2D Ising noise model

Ising model to represent the 2D noise process $\{Z_{i,j}\}$ with $i,j = 1,2,..,L$ (for $L \times L$ images).

Originally, in the Ising model, lattice variables are called spins $\{S_{i,j}\}$ and they are allowed to take only two opposite states: spin up ($s_{i,j} = +1$) or spin down ($s_{i,j} = -1$). In this case, the probability of a lattice configuration $\{s_{i,j}\}$ is provided by the Gibbs formula (Huang (1987)):

$$P(\mathbf{s}) \propto \exp\left(\beta \sum_{i,j} s_{i,j}\left(s_{i+1,j} + s_{i,j+1}\right) + H \sum_{i,j} s_{i,j}\right) \tag{26}$$

where $\mathbf{s}$ is a vector containing all the variables of the lattice $\{s_{i,j}\}$, β is called the interaction coefficient and H is the external magnetic field. The effect of the parameter β is to regulate the interaction among neighbor spins, for instance for $\beta \to 0$ the spins tend to be independent each other, on the other hand, if $|\beta|$ is higher than a critical value of $\beta_c = 0.44$, then the lattice is magnetized (a majority of the spins are in the same state) (Huang (1987)). On the other side, a positive (negative) parameter H induces spins to adopt the "+1" ("-1") state.

Since we want to model binary images we need to apply a mapping from the lattice with spin states to a new lattice with binary values "0" and "1" ($\{S_{i,j}\} \to \{Z_{i,j}\}$). For this mapping we consider the following relationship:

$$z_{i,j} = \frac{x_{i,j} + 1}{2} \tag{27}$$

Using the mapping (27), the equation (26) and after applying some algebraic operations we finally reach to the probability of a 2D Ising noise process which is:

$$P(\mathbf{z}) \propto \exp\left(4\beta n_{11} + 2(H - 4\beta)n_1\right) \tag{28}$$

where $\mathbf{z}$ is a vector containing all the variables of the lattice $\{z_{i,j}\}$ (also known as "pixels" in an image processing context), n_{11} is the number of horizontal and vertical pixel pairs where both pixels have a "1" value and n_1 is the total number of pixels with the value "1".

Following the reasoning of the previous subsections, the ML decoder for this case relies on the maximization of the following $g_I(.)$ function:

$$g_I(n_{11}, n_1) = 2\beta n_{11} + (H - 4\beta)n_1 \tag{29}$$

Note that different scenarios can take place depending on the values of parameters H and β, for example, if $\beta > 0$ and $H < 4\beta$, then the ML decoder will choose the code-word which produces the maximum n_{11} and, at the same time, the minimum n_1 within the set of vectors $\mathbf{z}^k = \mathbf{y} \oplus \mathbf{x}^k$, $\mathbf{x}^k \in \mathcal{C}$.

In order to simplify the equation (29), in this paper we provide an approximation which is based on the Bragg-Williams approximation already used in physics for the estimation of the critical temperature in the Ising model (Huang (1987)). The Bragg-Williams approximation states that

$$\frac{n_{11}}{2n} \approx \left(\frac{n_1}{n}\right)^2 \tag{30}$$

Replacing the approximation (30) in (29) we get a simpler approximated $\hat{g}_I(.)$ function:

$$\hat{g}_I(n_1) = \frac{n_1}{n}\left(\frac{n_1}{n} - \left(1 - \frac{H}{4\beta}\right)\right) \tag{31}$$

Notice that the function $\hat{g}_I(n_1)$ is quadratic and convex, therefore it has a unique minimum and the ML decoder has a similar behavior as the case of Polya contagion model, it means, the estimated input could be the closest or the farthest depending on the received codeword. A sketch of (31) is shown in Fig. 4.

IV CONCLUSIONS

Several noise models for a binary communication channel were analyzed. Bernoulli and Markov models were introduced for comparing them to the Polya contagion model previously analyzed in the bibliography. Also, a logarithmic distribution for the noise process was specially studied as it has conditional probabilities reaching to a constant value $\beta < 1$, which is different to the Polya contagion model. We demonstrated that for Markov chain and Logarithmic noise models, if some particular conditions are considered, the ML decoder is reduced to maximize or minimize the number of transitions q. Additionally, a two dimensional Ising model was analyzed since it is usually applied in image processing. We show that a ML decoding algorithm can be reduced to counting "0's" and "1's" when the Bragg-Williams approximation is applied to binary images.

In summary, this work provides new mathematical results that can be useful for the implementation of new decoders taking advantage of already known noise processes. One-dimensional noise models, like Markov and logarithmic cases here discussed, can be used for modeling burst like noise in a communication channel, where the probability of an error in a bit is dependent on the rest of the bits errors. On the other side, the 2D-Ising model developed in this paper, can be used directly for modeling spot like noise in black & white images, for example in digitally scanned images or scanned photocopies where the degradation of the image is not well modeled through an i.i.d. (independent identically distributed) variable associated to pixels. The Bragg-Williams approximation here introduced could be considered in more general Markov Random Fields. It will be discussed in a future work

Acknowledgements: C. F. Caiafa acknowledges the support of Facultad de Ingenieria, Universidad de Buenos Aires, Argentina (Beca Doctoral Peruilh). This work was partially supported by UBACyT I036 Project. The authors thank Eng. Facundo Caram for his useful comments on the first draft of this paper.

References

ALajaji, F. and Fuja T., “A Communication Channel modeled on contagion,” *IEEE Trans. on Inf. Theory*, **49**, 2035-2041 (1994)..

Barbero A., Ellingsen P., Spisante S. and Ytrehus O.,“Maximum Likelihood Decoding of Codes on the Z-channel” *IEEE International Conference on Communications, 2006. ICC ’06.* , Istanbul, 1200-1205 (2006).

Chi-Chao Chao, McEliece R.J., Swanson L., Rodemich E.R., “Performance of binary block codes at low signal-to-noise ratios ,”*IEEE Trans. Inf. Theory*, **38**, No. 6. pp. 1677-1687 (1992).

Douglas J. B., *Analysis with Standard Contagious Distributions*, International Co. Publishing House, (1980).

Feller W., *An Introduction to Probability Theory and Its Applications*, Volume 1, J. Wiley (1950).

Geman S. and Geman D., “Stochastics relaxation, Gibbs distributions, and the Bayesian restoration of images,”*IEEE Trans. Patt. Anal. Machine Intell.*, **6**, No. 6. pp. 721-741 (1984).

Greaffeath D., “Introduction to Random Fields ,”in *Denumerable Markov Chains*, New York: Springer-Verlag, pp. 425-458, 1976.

Haykin, S. *Communication Systems*, 4th Edition , J. Wiley (2001).

Huang, K. *Statistical Mechanics*, 4nd Edition , J. Wiley, New Jersey, (1987).

Hui Jin, McEliece R.J., “Coding theorems for turbo code ensembles ,”*IEEE Trans. Inf. Theory*, **48**, No. 6. pp. 1451-1461 (2002).

Khandekar A. and Mc Eliece R. J.,“On the Complexity of Reliable Communication on the Erasure Channel” *IEEE International Symposium on Information Theory, 2001. ISIT ’01.* , Washington, 7803-7123 (2001).

Moreira J.C. and Farrell P.G., *Essentials of Error-Control Coding*, 1st Edition , J. Wiley (2006).

Polya G. and Eggenberger F., “Über die Statistik Verketteter Vorgänge ” *Z. Angew. Math. Mech.* **3**, pp. 279-289 (1923).

Siromoney G., “The General Dirichlet’s Series Distribution ” *Journal of the Indian Statistical Association*, **2**, No. 2, (1964).