On the code reverse engineering problem

Mathieu Cluzeau, Jean-Pierre Tillich

On the Code Reverse Engineering Problem

Mathieu Cluzeau*, Jean-Pierre Tillich*
* INRIA, B.P. 105, 78153 Le Chesnay Cedex - France,
Email :{mathieu.cluzeau,jean-pierre.tillich}@inria.fr

***Abstract*— This article deals with the problem of quantifying how many noisy codewords have to be eavesdropped in order to reverse engineer a code. The main result of this paper is a lower bound on this quantity and the proof that this number is logarithmic in the length for LDPC codes.**

I. Introduction

The problem we address here is to be understood in the more general context of reverse-engineering a communication system. The general problem is, for an observer (or a spy), to recover the transmitted information from the knowledge of the observed stream. But he does not know anything about the characteristics of the different elements except the noisy channel, and so his first goal is to determine which elements have been used in the communication system. This problem arises for instance in a military context ; reverse engineering of the error correction component has been studied in [8], [1], [2] whereas reverse engineering of the scrambler has been considered in [3].

Here, we are interested in reverse engineering the error-correcting code C which has been used for communication. We call this the *CRE problem* (which stands for "Code Reverse Engineering"). We make the assumption that the observer knows that C has been chosen among a family $\mathcal{E}$ of codes of a given length n. Throughout the paper we will assume that C has been chosen uniformly at random among $\mathcal{E}$ and that M codewords $X^1, \ldots, X^M$ have also been chosen uniformly at random in C independently from each other and transmitted over the communication channel. We assume that the communication channel is a discrete memoryless channel. We denote by $Y^1, \ldots, Y^M$ the received words and set $\boldsymbol{X} \stackrel{\text{def}}{=} (X^i)_{1 \leq i \leq M}$ and $\boldsymbol{Y} \stackrel{\text{def}}{=} (Y^i)_{1 \leq i \leq M}$. We assume that the observer has observed all these noisy codewords Y^i's and wish to find which code has been chosen in $\mathcal{E}$. We also assume that all codes in $\mathcal{E}$ have the same length n and rate R.

We denote for a couple of discrete random variable X and Y their respective (binary) entropies by $H(X)$ and $H(Y)$, their mutual information by $I(X;Y)$ and the conditional entropy of X given Y by $H(X|Y)$.

II. A Capacity Formula for the CRE Problem

A. A Capacity Problem

As explained in the introduction, one of the fundamental issues which has to be addressed when trying to solve the CRE problem is to estimate the number of intercepted messages which is required to be able to find with good probability the code used during transmission. The answer of this question will be seen to heavily depend on two parameters of the code family $\mathcal{E}$ being used for transmission:
(i) its size $|\mathcal{E}|$,
(ii) a quantity related to its average error-correcting properties $\gamma(\mathcal{E})$.

We define this quantity here by

$$\gamma(E) = I(X;Y|C) \tag{1}$$

where C is chosen in $\mathcal{E}$ as explained above, X is chosen uniformly at random in C and Y is the received word after sending X through the discrete memoryless channel under consideration. This quantity lies between nR and 0 and is close to nR when most of the codes in $\mathcal{E}$ enable us to recover X from Y most of the time.

The issue of giving lower and upper bounds on the size of M which is required for recovering C with good probability is in essence an estimation of a channel capacity. The channel that the spy sees can be viewed as follows. The input alphabet of the channel is $\mathcal{E}$ and the output alphabet of the channel is the set $\mathcal{B}$ of all possible output vectors for the Y_i's. In this case, he also knows that that the repetition code has been used (C was transmitted M times). The channel is then a discrete memoryless channel, when an input symbol C is transmitted (*i.e.* a code in $\mathcal{E}$) a random word X is chosen in C and transmitted through the real communication channel to become an element Y of $\mathcal{B}$. The crucial fact is that for any $y_1, \ldots, y_M \in \mathcal{B}$ and any $c \in \mathcal{E}$ we have

$$\mathbf{P}(\boldsymbol{Y} = (y_1, \ldots, y_M) | C = c) = \prod_{i=1}^{M} \mathbf{P}(Y_i = y_i | C = c). \tag{2}$$

From now on, we will use the following convention:
Notation: X and Y denote random variables as defined above.

Viewing the CRE problem in this way motivates to look at the mutual information between Y and C. There is a very simple formula for this quantity involving $\gamma(\mathcal{E})$, namely

Lemma 1: We have

$$I(Y;C) = I(X;Y) - \gamma(\mathcal{E}).$$

Proof: This is basically a consequence of the fact that the triple (C, X, Y) forms a Markov chain (the conditional distribution of Y depends only on X and is conditionally independent of C). From this, the reverse triple (Y, X, C) forms a Markov chain too. Let us observe now that $I(Y; C, X)$ can be expressed in two different ways

$$\begin{aligned} I(Y;C,X) &= I(Y;C) + I(Y;X|C) \\ &= I(Y;X) + I(Y;C|X). \end{aligned}$$

From the Markov chain property $I(Y;C|X)=0$. We deduce the following expression for $I(Y;C)$:

$$I(Y;C)=I(Y;X)-I(Y;X|C)=I(X;Y)-\gamma(\mathcal{E}).$$

■

B. A General Lower Bound on the Required Number of Intercepted Messages

The conditional entropy $H(C|\boldsymbol{Y})$ can be related to $\gamma(\mathcal{E}), M, n, I(X;Y)$ and $|\mathcal{E}|$ by the following proposition

Proposition 1:

$$H(C|\boldsymbol{Y})\geq \log_2(|\mathcal{E}|)-M\left(I(X;Y)-\gamma(\mathcal{E})\right).$$

Proof: Let us first write that

$$H(C|\boldsymbol{Y})=H(C)-I(\boldsymbol{Y};C)=\log_2|\mathcal{E}|-I(\boldsymbol{Y};C). \quad (3)$$

>From the independence property (2) we know that $H(\boldsymbol{Y}|C)=\sum_{i=1}^{M}H(Y^i|C)$. Since $I(\boldsymbol{Y};C)=H(\boldsymbol{Y})-H(\boldsymbol{Y}|C)$, we obtain

$$\begin{aligned} I(\boldsymbol{Y};C) &= H(\boldsymbol{Y})-\sum_{i=1}^{M}H(Y^i|C) \\ &\leq \sum_{i=1}^{M}H(Y^i))-\sum_{i=1}^{M}H(Y^i|C) \\ &\leq \sum_{i=1}^{M}I(Y^i;C). \end{aligned} \quad (4)$$

From Equations (3) and (4) we deduce the proposition above.

■

This proposition gives a lower bound on the number of messages which have to be intercepted in order to have a vanishing error probability when choosing the most likely C. Indeed, by denoting this probability by P_e and by using Fano's inequality, we obtain

$$h(P_e)+P_e\log_2(|\mathcal{E}|-1)\geq H(C|\boldsymbol{Y}),$$

where h stands for the binary entropy function, *i.e.*

$$h(x)\stackrel{\text{def}}{=}-x\log_2-(1-x)\log_2(1-x).$$

We deduce from this that

$$\frac{h(P_e)}{\log_2(|\mathcal{E}|-1)}+P_e\geq\frac{\log_2|\mathcal{E}|}{\log_2(|\mathcal{E}|-1)}-M\frac{I(X;Y)-\gamma(\mathcal{E})}{\log_2(|\mathcal{E}|-1)}.$$

In other words, if we want to recover C from $\boldsymbol{Y}$ with vanishing error probability when n goes to infinity and if $|\mathcal{E}|$ goes to infinity with it, we need M to be at least of size $\big(1+o(1)\big)m(\mathcal{E})$ where

$$m(\mathcal{E})\stackrel{\text{def}}{=}\frac{\log_2|\mathcal{E}|}{I(X;Y)-\gamma(\mathcal{E})}.$$

III. The case of linear codes of a given length and dimension

Let us consider the case where $\mathcal{E}$ is the set of linear codes over $\mathbb{F}_2$ of length n and dimension k. The cardinality $|\mathcal{E}|$ is given by $\left[\begin{array}{c} n \\ k \end{array}\right]$, which is is the number of vector subspaces of dimension k of a vector space of dimension n over the binary field. It can be written as

$$\left[\begin{array}{c} n \\ k \end{array}\right]=\prod_{i=0}^{k-1}\frac{2^{n-i}-1}{2^{k-i}-1}.$$

Let us notice that $\left[\begin{array}{c} n \\ k \end{array}\right]=2^{(n-k)k}\prod_{i=0}^{k-1}\frac{1-2^{i-n}}{1-2^{i-k}}$. By using the fact that $\frac{1}{\prod_{i=1}^{\infty}1-2^{-i}}\leq 4$ we finally obtain $2^{(n-k)k}\leq\left[\begin{array}{c} n \\ k \end{array}\right]\leq 2^{(n-k)k+2}$, from which we deduce

$$\log_2(|\mathcal{E}|)\sim k(n-k), \quad (5)$$

as $k(n-k)$ tends to infinity.

Here, we have used the following notation

Notation 1:

$$f(x)\sim g(x)$$

as x goes to infinity, means that

$$\lim_{x\to\infty}f(x)/g(x)=1.$$

Concerning the other terms which are involved in the definition of $m(\mathcal{E})$, let us first observe that the distribution of X satisfies

$$\mathbf{P}(X=0)=\frac{1}{2^k},\ \mathbf{P}(X=x)=\frac{1-2^{-k}}{2^n-1}\ \text{ for } x\neq 0.$$

If we bring in a random variable U uniformly distributed on $\mathbb{F}_2^n$ and if we let V be its corresponding output for the communication channel, it is readily checked that as n goes to infinity

$$I(X;Y)\sim I(U;V).$$

The term appearing on the right side can be rewritten as

$$I(U;V) \quad = \quad n\mathbf{Cap}_{\mathbf{U}},$$

where

Definition 1: $\mathbf{Cap}_{\mathbf{U}}$ is the mutual information between a random variable uniformly distributed on $\mathbb{F}_2$ and the corresponding output from the channel.

We finally obtain

$$I(X;Y)\sim n\mathbf{Cap}_{\mathbf{U}} \quad (6)$$

as n tends to infinity.

Turning to the problem of estimating $\gamma(\mathcal{E})$, it is readily checked that classical arguments used in the proof of the direct part of Shannon's capacity theorem allow to show that for any $\varepsilon>0$, most of linear codes of rate strictly below $\mathbf{Cap}_{\mathbf{U}}$ have probability of error after decoding which is smaller than ε for n large enough. This can be used to show that $H(X|Y,C)=o(n)$ for a fixed rate R strictly below $\mathbf{Cap}_{\mathbf{U}}$. From this we deduce that under this condition

Lemma 2: Let $R < \mathbf{Cap_U}$. If $\mathcal{E}$ is the family of linear codes of rate R and length n, then $\gamma(\mathcal{E}) \sim nR$, as n goes to infinity.

Putting all these facts together we deduce that

Proposition 2: Let $R < \mathbf{Cap_U}$. If $\mathcal{E}$ is the family of linear codes of dimension k, length n and rate R, then

$$m(\mathcal{E}) \sim k \frac{1-R}{\mathbf{Cap_U} - R}$$

as n goes to infinity.

Notice that when there are no errors, then our bound claims that we need at least k intercepted words to reconstruct our code, which was to be expected.

IV. LDPC Codes

An interesting example which might arise in this setting is the reverse engineering of LDPC codes. To simplify the discussion we will assume in the whole section that the channel is a binary symmetric channel with crossover probability p and that the parity checks of the LDPC code family have all constant weight t. As we will see, this already captures interesting features of the problem and avoids more general but also much more complicated statements.

A. A Single Parity Check Code

A first toy example whose importance will become apparent in Section V corresponds to the case where $\mathcal{E}$ consists of all codes of a given length n whose parity check matrix consists of a single parity-check of weight t.

From the definition of our set $\mathcal{E}$, we have $|\mathcal{E}| = \binom{n}{t}$. Let us compute $I(X;Y|C)$. We have:

$$I(X;Y|C) = H(X|C) - H(X|Y,C) = (n-1) - H(X|Y,C).$$

Let us calculate now $H(X|Y,C)$. This last quantity does not depend on C. Without loss of generality we may assume that C is the code where the t first positions belong to the parity-check equation of the code. We assume that X' is chosen uniformly in this code, that Y' is its corresponding output from the channel. We split now the support of our words into two parts, one part for the support of the parity equation defining the code and the other part for the rest of the positions. We let X'_1 and Y'_1 be the first part of X' and Y' respectively. We denote by X'_2 and Y'_2 the last part of X' and Y' respectively.

$$H(X|Y,C) = H(X'|Y') = H(X'_1|Y'_1) + H(X'_2|Y'_2).$$

Notice that $H(X'_2|Y'_2) = (n–t)h(p)$. For the first part, we write

$$\begin{aligned} H(X'_1|Y'_1) &= H(X'_1) - H(Y'_1) + H(Y'_1|X'_1) \\ &= (t-1) - H(Y'_1) + th(p). \end{aligned}$$

For computing $H(Y'_1)$, we may observe that, for any $y'_1 \in \mathbb{F}_2^t$, the value of $\mathbf{P}(Y'_1 = y'_1)$ only depends on the parity of $|y'_1|$.

Let A_0 (resp. A_1) denote the event: the parity of $|y_t|$ is even (resp. odd). Then

$$\begin{aligned} H(Y'_1) &= \mathbf{P}(A_0) \log\left(\frac{2^{t-1}}{\mathbf{P}(A_0)}\right) + \mathbf{P}(A_1) \log\left(\frac{2^{t-1}}{\mathbf{P}(A_1)}\right) \\ &= (t-1) + h\left(\mathbf{P}(A_0)\right). \end{aligned}$$

Note that $\mathbf{P}(A_0) = \frac{1+(1-2p)^t}{2}$. This quantity will arise often in what follows and we denote it by

Notation 2:

$$p_t \stackrel{\text{def}}{=} \frac{1+(1-2p)^t}{2}.$$

Putting all these calculations together, we obtain

$$\begin{aligned} H(X|Y,C) &= nh(p) - h\left(p_t\right) \\ I(X;Y) - I(X;Y|C) &= 1 + o(1) - h\left(p_t\right). \end{aligned}$$

The reason of the $o(1)$ term comes from the fact that the distribution of X is not completely uniform: it can be proven that $I(X;Y) = n(1-h(p)) + o(1)$.

Finally, we have

$$m(\mathcal{E}) = \frac{\log_2 \binom{n}{t}}{1 + o(1) - h\left(p_t\right)},$$

from which we deduce

Proposition 3: For fixed t:

$$m(\mathcal{E}) \sim \frac{t \log_2 n}{1 - h\left(p_t\right)},$$

as n tends to infinity.

B. The Juxtaposition of Single Parity-Check Codes of Size t

We consider now the case where n is a multiple of t and where $\mathcal{E}$ is the set of all codes of length n which are the juxtaposition of n/t single parity-check codes of size t. These codes have parity-check matrices with n/t rows with constant row weight t and column weight 1.

Concerning the cardinality of this ensemble of codes, we have

$$|\mathcal{E}| = \frac{n!}{(t!)^{\frac{n}{t}} (\frac{n}{t})!}.$$

By using the Stirling formula we obtain for n going to infinity

$$\begin{aligned} \log_2(|\mathcal{E}|) &= \big(n + o(n)\big)\big(\log_2 n - \log_2 t - \frac{\log_2 n}{t}\big) \\ &\sim n(1 - 1/t) \log_2 n. \end{aligned}$$

Let us compute now $I(Y;X|C)$. We first write

$$I(Y;X|C) = H(X|C) - H(X|Y,C) = \Big(n - \frac{n}{t}\Big) - H(X|Y,C).$$

As in the previous case, we will decompose X and Y, and this time we will split the support into $\frac{n}{t}$ parts corresponding to the decomposition of C into single parity-check codes of size t.

By performing similar calculations as in the previous subsection we obtain

$$H(X|Y,C) = \frac{n}{t}\left(th(p) - h\left(p_t\right)\right),$$

and deduce

Proposition 4: For n going to infinity

$$m(\mathcal{E}) \sim \frac{(t-1)\log_2 n}{1 - h\left(p_t\right)}. \tag{7}$$

C. Regular LDPC Codes

In this subsection for the sake of simplicity, we consider the case of regular LDPC codes: all parity check equations have weight t and all code positions are involved in exactly s parity checks. We assume that n is a multiple of t and that the parity-check matrices of the codes in $\mathcal{E}$ are the set of binary matrices of row weight t and column weight s.

Recall that all these codes can be obtained by specifying their Tanner graph in the following way. Let $r \stackrel{\text{def}}{=} \frac{ns}{t}$. This is the number of rows of the low-density parity check matrices of the codes in $\mathcal{E}$. We construct the Tanner graph with a bipartite graph between n variable nodes and r check nodes by:
(i) attaching to each variable node s sockets and to each check node t sockets,
(ii) choosing a permutation on sn elements which specifies a matching between the sn sockets attached to the variable nodes and the same number of sockets attached to the check nodes ;
(iii) this matching specifies a (multi)graph between the n variable nodes and the r check nodes.

All parity-check matrices with constant row and column weight are associated to a graph built in this way. It might happen that some of these multi-graphs do not specify a Tanner graph. This comes from the fact that this construction does not avoid multiple edges. However, it is straightforward to show that at least a constant fraction of such multi-graphs are admissible Tanner graphs. A same code is associated to several Tanner graphs obtained in this way: all $r!$ permutations of the rows of the parity-check matrix specify the same code. However, for fixed t and n going to infinity, the proportion of codes which are attached to more than $r!$ different parity-check matrices of the aforementioned form goes to 0. This is related to the fact that as n tends to infinity most of the dual of these codes contain exactly r codewords of weight t. All this implies that as n goes to infinity $\log_2(|\mathcal{E}|) \sim \log_2((sn)!/r!)$, and this can be simplified to yield

$$\log_2(|\mathcal{E}|) \sim \frac{s(t-1)}{t} \log_2 n. \tag{8}$$

As before $I(X;Y) = n(1-h(p))+o(1)$, but the calculation of $I(X;Y|C)$ is much more involved. For instance, the threshold p_0 which is defined as the supremum of the p's for which $\lim_{n\to\infty} \frac{I(X;Y|C)}{nR} = 1$ is not known exactly. R stands here for the designed rate of the LDPC code family, that is $R \stackrel{\text{def}}{=} 1 - s/t$. Only lower and upper bounds are known for this quantity [4], [6].

Proposition 5: For $p < p_0$, we have as n tends to infinity

$$m(\mathcal{E}) \sim \frac{s(1-1/t)}{1-h(p)-R} \log_2 n. \tag{9}$$

V. An Algorithm for Determining C

Here, we will present an algorithm for determining C from the noisy codewords Y^i that have been received by finding words of weight t in the dual code. We give this algorithm for a binary symmetric channel[1] with crossover probability p. This algorithm is based on the fact that, if h belongs to the dual code $C^\perp$ and if y denotes the received codeword then:

$$\mathbf{P}\left(< h, y >= 0\right) = p_t.$$

And, of course, this probability is equal to $\frac{1}{2}$ if h does not belongs to $C^\perp$. The algorithm for recovering C consists in testing all parity-checks of weight t and detecting which ones belong to the dual code. For deciding that a given parity-check belongs to $C^\perp$, we perform a statistical test with a threshold. If the number of Y^i's such that $< h, Y^i >= 1$ is less than the threshold then we decide that h belongs to the dual code ; otherwise we decide that h does not belong to the dual code.

Algorithm:
inputs: $\boldsymbol{Y} = (Y^1 \cdots Y^M)$, a weight t.
output: The dual code of C or a subcode of the dual code.

1. For every h of weight t , compute $|h\boldsymbol{Y}|$
   If $|h\boldsymbol{Y}| \leq T$, then decide that $h \in C^\perp$
2. Return all such h's.

The value of the threshold T is chosen according to t and t is chosen as small as possible. To analyze how the value of T affects the number of bad candidates (*i.e.* the h's which do not belong to $C^\perp$) returned by the algorithm let us bound their expected number $\mathrm{E}_{\mathrm{BAD}}$.

$$\mathrm{E}_{\mathrm{BAD}} \leq \binom{n}{t} \sum_{i=0}^{T} \binom{M}{i} \frac{1}{2^M} \leq \binom{n}{t} 2^{-M\left(1-h\left(\frac{T}{M}\right)\right)}. \tag{10}$$

How the input value t is chosen depends on the family $\mathcal{E}$. We will consider several cases.

A. The Family of Single Parity-Check Codes

For this family given in Subsection IV-A, the value chosen for t corresponds to the size of the parity-check equation defining the family. If we want that the probability of accepting the right parity-check equation goes to 1 with the length n we may choose T such that

$$T = M(1-p_t) + M^{2/3}.$$

In this case, choosing M of the same order as $m(\mathcal{E})$, that is

$$M = \frac{(t+\varepsilon)\log_2(n)}{1-h\left(p_t\right)},$$

for an arbitrary small value of ε, yields the following upper-bound on $\mathrm{E}_{\mathrm{BAD}}$:

$$\begin{aligned} \mathrm{E}_{\mathrm{BAD}} &\leq \binom{n}{t} 2^{-M\left(1-h\left(\frac{T}{M}\right)\right)} \\ &\leq n^t 2^{-\alpha(t+\varepsilon)\log_2 n} \\ &\leq n^{t(1-\alpha)-\alpha\varepsilon}, \end{aligned}$$

[1] It can be generalized to other channels but we give it here for this channel to simplify the discussion.

with $\alpha = \frac{1-h(p_t-M^{-1/3})}{1-h(p_t)}$. This shows that as the length goes to infinity, the probability of having bad candidates goes to zero. In this case, the lower bound on the number of messages which follows from the application of Propositions 1 and 3 is tight.

B. The Family of Regular LDPC Codes

The input value for t in the previous algorithm is chosen again as the size of the parity-check equations defining the family. There are $r = \frac{ns}{t}$ dual codewords of weight t that our algorithm has to detect. T is chosen in such a way that both the expected number $\mathrm{E_{UND}}$ of undetected dual codewords of weight t and the expected number of wrongly detected codewords (*i.e.* $\mathrm{E_{BAD}}$) go to zero as the length n goes to infinity.

Let us first bring in a few useful quantities: $\varepsilon_t \stackrel{\text{def}}{=} p_t - \frac{1}{2} = (1-2p)^t/2$, $\lambda \stackrel{\text{def}}{=} \frac{M}{\log_2 n}$. We choose T of the form

$$T = M\left(1 - p_t + \varepsilon\right),$$

for some ε which will be specified later on. Let us notice that by using Chernoff's inequality we have

$$\mathrm{E_{UND}} \leq r 2^{-2M\varepsilon^2} \leq n^{1-2\varepsilon^2\lambda}.$$

On the other hand by using the inequality $\binom{n}{t} \leq n^t$ in (10) we obtain

$$\begin{aligned} \mathrm{E_{BAD}} &\leq n^t 2^{-M\left(1-h(1/2-(\varepsilon_t-\varepsilon))\right)} \\ &\leq n^{t-\lambda(1-h(1/2-(\varepsilon_t-\varepsilon)))}. \end{aligned}$$

We are therefore looking for λ and ε which satisfy simultaneously

$$\begin{aligned} 1 - 2\varepsilon^2\lambda &< 0 \\ t - \lambda(1 - h(1/2 - (\varepsilon_t - \varepsilon))) &< 0. \end{aligned}$$

For any $0 < \varepsilon < \varepsilon_t$, a value of λ above $\max\left(\frac{1}{2\varepsilon^2}, \frac{t}{1-h(1/2-(\varepsilon_t-\varepsilon))}\right)$ does the job. By using the inequality $1 - h(1/2 - x) \geq \frac{2x^2}{\ln 2}$ we see that a value above $\frac{1}{2}\max\left(\frac{1}{\varepsilon^2}, \frac{t\ln 2}{(\varepsilon_t-\varepsilon)^2}\right)$ is acceptable. We minimize this quantity by choosing $\varepsilon = \varepsilon_t \frac{\sqrt{t\ln 2}-1}{t\ln 2-1}$, and it can be checked by straightforward calculations that we can choose M of the form $\lambda(t)\log_2 n$ with $\lambda(t) \sim \frac{t}{1-h(p_t)}$ as t tends to infinity. This is asymptotically the same quantity as in the previous example. This time, it does not meet the lower bound $m(\varepsilon)$ of Proposition 5. However, it captures the logarithmic behavior of this quantity and shows:
(i) that a logarithmic number of codewords is necessary and sufficient for recovering this LDPC code family,
(ii) that this can be achieved efficiently in polynomial time when t is fixed.

C. The Family of Linear Codes of a Given Rate

In this case, a good choice for t in the previous algorithm corresponds to choose values slightly above the Gilbert-Varshamov distance $d_{\mathrm{GV}}^{\perp} = \lceil nh^{-1}(R) \rceil$ of the dual code. The point is that it is the smallest value for which most linear codes of rate R, have a basis of $C^{\perp}$ formed only by words of weight t. This can verified by standard probabilistic calculations.

However in this case, even by keeping only a constant fraction of dual codewords of weight t by choosing for instance the threshold T as

$$T = M(1 - p_t),$$

a vanishing expectation $\mathrm{E_{BAD}}$ is only attained for M at least of order

$$M \sim \frac{nh^{-1}(R)}{1 - h\left(\frac{1-(1-2p)^{nh^{-1}(R)}}{2}\right)}.$$

This quantity has unfortunately an exponential behavior in n:

$$\frac{nh^{-1}(R)}{1 - h\left(\frac{1-(1-2p)^{nh^{-1}(R)}}{2}\right)} \sim \frac{2\ln 2 h^{-1}(R) n}{(1-2p)^{2nh^{-1}(R)}}.$$

The algorithm presented here does not achieve the goal of recovering the right code with a linear number of codewords.

VI. Conclusion

A logarithmic number of codewords is necessary and sufficient to reverse engineer an LDPC code. Moreover this task can be achieved in polynomial time. However, it is not clear how we could improve the algorithm presented here to achieve with polynomial time complexity the reverse engineering of such a code family by using less codewords. A challenging task would be for instance to be able to reverse engineer in polynomial time an LDPC code family with parity-check equations of weight t by using only (asymptotically in t) of order $\frac{t-1}{1-h(p_t)}$ codewords instead of $\frac{t}{1-h(p_t)}$. This would match the lower bound for reverse engineering the juxtaposition of single parity-check codes of size t. A possible way to approach this issue would be to assign probabilities that a parity-check equation of weight t belongs to the dual of the code together with Gallager's decoding algorithm in the spirit of [2]. For the linear code family it is unclear if the linear lower bound provided by Proposition 1 is tight or not.

References

[1] C. Chabot, "Recognition of a code in a noisy environment," *IEEE Conference, ISIT'07*, pp. 2210–2215, 2007;
[2] M. Cluzeau, "Block code reconstruction using iterative decoding techniques," *IEEE Conference, ISIT'06*, pp. 2269–2273, 2006;
[3] M. Cluzeau, "Reconstruction of a linear scrambler," *IEEE Transactions on Computers*, 2007;
[4] R. G. Gallager, "Low Density Parity Check Codes," *MIT Press*, 1963;
[5] E.N. Gilbert, "A comparison of signaling alphabets," *Bell Syst. Tech. J.*,vol. 31, pp. 504–522, 1952;
[6] A. Montanari, "Tight bounds for LDPC and LDGM codes under MAP decoding", *IEEE Transactions on Information Theory*, vol. 51, pp. 3221–3246, 2005;
[7] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 1948; pt. II, pp. 623–656, 1948.
[8] A. Valembois, "Detection and recognition of a binary linear code," *Discrete Applied Mathematics*, vol. 111(1-2), pp. 199–218,2001;
[9] R.R. Varshamov, "Estimate of the number of signals in error correcting codes," *Dokl. Acad. Nauk,*, vol. 117, pp. 739–741, 1959.