Statistical Decoding

Thomas Debris-Alazard, Jean-Pierre Tillich

Statistical Decoding

Thomas Debris-Alazard * Jean-Pierre Tillich *

March 31, 2017

Abstract

The security of code-based cryptography relies primarily on the hardness of generic decoding with linear codes. The best generic decoding algorithms are all improvements of an old algorithm due to Prange: they are known under the name of information set decoding techniques (ISD). A while ago a generic decoding algorithm which does not belong to this family was proposed: statistical decoding. It is a randomized algorithm that requires the computation of a large set of parity-check equations of moderate weight. We solve here several open problems related to this decoding algorithm. We give in particular the asymptotic complexity of this algorithm, give a rather efficient way of computing the parity-check equations needed for it inspired by ISD techniques and give a lower bound on its complexity showing that when it comes to decoding on the Gilbert-Varshamov bound it can never be better than Prange's algorithm.

1 Introduction

Code-based cryptography relies crucially on the hardness of decoding generic linear codes. This problem has been studied for a long time and despite many efforts on this issue [Pra62, Ste88, Dum91, Bar97, MMT11, BJMM12, MO15] the best algorithms for solving this problem [BJMM12, MO15] are exponential in the number of errors that have to be corrected: correcting t errors in a binary linear code of length n has with the aforementioned algorithms a cost of $2^{ct(1+o(1))}$ where c is a constant depending of the code rate R and the algorithm. All the efforts that have been spent on this problem have only managed to decrease slightly this exponent c. Let us emphasize that this exponent is the key for estimating the security level of any code-based cryptosystem.

All the aforementioned algorithms can be viewed as a refinement of the original Prange algorithm [Pra62] and are actually all referred to as ISD algorithms. There is however an algorithm that does not rely at all on Prange's idea and does not belong to the ISD family: statistical decoding proposed first by Al Jabri in [Jab01] and improved a little bit by Overbeck in [Ove06]. Later on, [FKI07] proposed an iterative version of this algorithm. It is essentially a two-stage algorithm, the first step consisting in computing an exponentially large number of parity-check equations of the smallest possible weight w, and then from these parity-check equations the error is recovered by some kind of majority voting based on these parity-check equations.

*Inria, SECRET Project, 2 Rue Simone Iff 75012 Paris Cedex, France, Email: {thomas.debris,jean-pierre.tillich}@inria.fr. Part of this work was supported by the Commission of the European Communities through the Horizon 2020 program under project number 645622 PQCRYPTO.

However, even if the study made by R. Overbeck in [Ove06] lead to the conclusion that this algorithm did not allow better attacks on the cryptosystems he considered, he did not propose an asymptotic formula of its complexity that would have allowed to conduct a systematic study of the performances of this algorithm. Such an asymptotic formula has been proposed in [FKI07] through a simplified analysis of statistical decoding, but as we will see this analysis does not capture accurately the complexity of statistical decoding. Moreover both papers did not assess in general the complexity of the first step of the algorithm which consists in computing a large set of parity-check equations of moderate weight.

The primary purpose of this paper is to clarify this matter by giving three results. First, we give a rigorous asymptotic study of the exponent c of statistical decoding by relying on asymptotic formulas for Krawtchouk polynomials [IS98]. The number of equations which are needed for this method turns out to be remarkably simple for a large set of parameters. In Theorem 2 we prove that the number of parity check equations of weight ωn that are needed in a code of length n to decode τn errors is of order $O(2^{n(H(\omega)+H(\tau)-1)})$ (when we ignore polynomial factors) and this as soon as $\omega \geq \frac{1}{2} - \sqrt{\tau - \tau^2}$. For instance, when we consider the hardest instances of the decoding problem which correspond to the case where the number of errors is equal to the Gilbert-Varshamov bound, then essentially our results indicate that we have to take *all* possible parity-checks of a given weight (when the code is assumed to be random) to perform statistical decoding. This asymptotic study also allows to conclude that the modeling of iterative statistical decoding made in [FKI07] is too optimistic. Second, inspired by ISD techniques, we propose a rather efficient method for computing a huge set of parity-check equations of rather low weight. Finally, we give a lower bound on the complexity of this algorithm that shows that it can not improve upon Prange's algorithm for the hardest instances of decoding.

This lower bound follows by observing that the number P_w of the parity-check equations of weight w that are needed for the second step of the algorithm is clearly a lower-bound on the complexity of statistical decoding. What we actually prove in the last part of the paper is that irrelevant of the way we obtain these parity-check equations in the first step, the lower bound on the complexity of statistical decoding coming from the infimum of these P_w's is always larger than the complexity of the Prange algorithm for the hardest instances of decoding.

2 Notation

As our study will be asymptotic, we neglect polynomial factors and use the following notation:

Notation 1. Let $f, g : \mathbb{N} \to \mathbb{R}$, we write $f = \tilde{O}(g)$ iff there exists a polynomial P such that $f = O(Pg)$.

Moreover, we will often use the classical result $\binom{n}{w} = \tilde{O}\left(2^{nH\left(\frac{w}{n}\right)}\right)$ where H denotes the binary entropy. We will also have to deal with complex numbers and follow the convention of the article [IS98] we use here: $\mathbf{i}$ is the imaginary unit satisfying the equation $\mathbf{i}^2 = -1$, $\Re(z)$ is the real part of the complex number z and we choose the branch of the complex logarithm with

$$\ln(z) = \ln|z| + \mathbf{i}\arg(z), \quad z \in \mathbb{C} \setminus [-\infty, 0],$$

and $\arg(z) \in [-\pi, \pi)$.

3 Statistical Decoding

In the whole paper we consider the computational decoding problem which we define as follows:

Problem 1. *Given a binary linear code of length n of rate R, a word $y \in \mathbb{F}_2^n$ at distance t from the code, find a codeword x such that $d_H(x, y) = t$ where d_H denotes the Hamming distance.*

Generally we will specify the code by an arbitrary generator matrix G and we will denote by $\mathrm{CSD}(G, t, y)$ a specific instance of this problem. We will be interested as is standard in cryptography in the case where $G \in \mathbb{F}_2^{Rn \times n}$ *is supposed to be random.*

The idea behind statistical decoding may be described as follows. We first compute a very large set $\mathcal{S}$ of parity-check equations of some weight w and compute all scalar products $\langle y, h\rangle$ (scalar product is modulo 2) for $h \in \mathcal{S}$. It turns out that if we consider only the parity-checks involving a given code position i the scalar products have a probability of being equal to 1 which depends whether there is an error in this position or not. Therefore counting the number of times when $\langle y, h\rangle = 1$ allows to recover the error in this position.

Let us analyze now this algorithm more precisely. To make this analysis tractable we will need to make a few simplifying assumptions. The first one we make is the same as the one made by R. Overbeck in [Ove06], namely that

Assumption 1. The distribution of the $\langle y, h\rangle$'s when h is drawn uniformly at random from the dual codewords of weight w is approximated by the distribution of $\langle y, h\rangle$ when h is drawn uniformly at random among the words of weight w.

A much simpler model is given in [FKI07] and is based on modeling the distribution of the $\langle y, h\rangle$'s as the distribution of $\langle y, h\rangle$ where the coordinates of h are i.i.d. and distributed as a Bernoulli variable of parameter w/n. This presents the advantage of making the analysis of statistical decoding much simpler and allows to analyze more refined versions of statistical decoding. However as we will show, this is an oversimplification and leads to an over-optimistic estimation of the complexity of statistical decoding. The following notation will be useful.

Notation 2.
· $S_w \stackrel{\triangle}{=} \{x \in \mathbb{F}_2^n : w_H(x) = w\}$ denotes the set of binary of words of length n of weight w;
· $S_{w,i} \stackrel{\triangle}{=} \{x \in S_w : x_i = 1\}$;
· $\mathcal{H}_w \stackrel{\triangle}{=} \mathscr{C}^\perp \cap S_w$;
· $\mathcal{H}_{w,i} \stackrel{\triangle}{=} \mathscr{C}^\perp \cap S_{w,i}$;
· $X \sim \mathscr{B}(p)$ means that X follows a Bernoulli law of parameter p ;
· $h \sim S_{w,i}$ means we pick h uniformly at random in $S_{w,i}$.

3.1 Bias in the parity-check sum distribution

We start the analysis of statistical decoding by computing the following probabilities which approximate the true probabilities we are interested in (which correspond to choosing h uniformly at random in $\mathcal{H}_{w,i}$ and not in $S_{w,i}$) under Assumption 1

$$q_1(e, w, i) = \mathbb{P}_{h \sim S_{w,i}}\left(\langle e, h\rangle = 1\right) \text{ when } e_i = 1$$

$$q_0(e, w, i) = \mathbb{P}_{h \sim S_{w,i}}\left(\langle e, h\rangle = 1\right) \text{ when } e_i = 0$$

These probabilities are readily seen to be equal to

$$q_1(e,w,i) = \frac{\sum\limits_{j \text{ even}}^{w-1} \binom{t-1}{j}\binom{n-t}{w-1-j}}{\binom{n-1}{w-1}}$$

$$q_0(e,w,i) = \frac{\sum\limits_{j \text{ odd}}^{w-1} \binom{t}{j}\binom{n-t-1}{w-1-j}}{\binom{n-1}{w-1}}$$

They are independent of the error and the position i. So, in the following we will use the notation q_1 and q_0. We will define the biases ε_0 and ε_1 of statistical decoding by

$$q_0 = \frac{1}{2} + \varepsilon_0 \ ; \ q_1 = \frac{1}{2} + \varepsilon_1$$

It will turn out, and this is essential, that $\varepsilon_0 \neq \varepsilon_1$. We can use these biases "as a distinguisher". They are at the heart of statistical decoding. Statistical decoding is nothing but a statistical hypothesis testing algorithm distinguishing between two hypotheses :

$$\mathscr{H}_0 \ : \ e_i = 0 \quad ; \quad \mathscr{H}_1 \ : \ e_1 = 1$$

based on computing the random variable V_m for m uniform and independent draws of vectors in $\mathcal{H}_{w,i}$:

$$V_m = \sum_{k=1}^{m} \operatorname{sgn}(\varepsilon_1 - \varepsilon_0) \cdot \langle y, h^k \rangle \in \mathbb{Z}$$

We have $\langle y, h^k \rangle \sim \mathscr{B}(1/2 + \varepsilon_l)$ according to $\mathscr{H}_l$. So the expectation of V_m is given under $\mathscr{H}_l$ by:

$$E_l = m \operatorname{sgn}(\varepsilon_1 - \varepsilon_0)(1/2 + \varepsilon_l)$$

We point out that we have $E_1 > E_0$ regardless of the term $\operatorname{sgn}(\varepsilon_1 - \varepsilon_0)$. In order to apply the following proposition, we make the following assumption:

Assumption 2. $\langle y, h^k \rangle$ are independent variables.

Proposition 1 (Chernoff's Bound)**.** *Let $0 < p < 1$, $Y_1, \cdots, Y_m$ i.i.d $\sim \mathscr{B}(p)$ and we set $Z_m = \sum_{k=1}^{m} Y_k$. Then,*

$$\forall t \geq 0, \quad \mathbb{P}\left(|Z_m - mp| \geq m\delta\right) \leq 2e^{-2m\delta^2}$$

Consequences: Under $\mathscr{H}_l$, we have

$$\mathbb{P}\left(|V_m - m \operatorname{sgn}(\varepsilon_1 - \varepsilon_0) \cdot (1/2 + \varepsilon_l)| \geq m \cdot \frac{|\varepsilon_1 - \varepsilon_0|}{2}\right) \leq 2 \cdot 2^{-m \cdot \frac{(\varepsilon_1 - \varepsilon_0)^2}{2\ln(2)}}$$

To take our decision we proceed as follows: if $V_m < \frac{E_0 + E_1}{2}$ where

$$\frac{E_1 + E_0}{2} = \frac{m}{2} \operatorname{sgn}(\varepsilon_1 - \varepsilon_0)(1 + \varepsilon_1 + \varepsilon_0)$$

we choose $\mathscr{H}_0$ and $\mathscr{H}_1$ if not. For the cases of interest to us (namely w and t linear in n) the bias $\varepsilon_1 - \varepsilon_0$ is an exponentially small function of the codelength n and it is obviously enough to choose m to be of order $O\left(\frac{\log n}{(\varepsilon_1-\varepsilon_0)^2}\right)$ to be able to make the good decisions on all n positions simultaneously.

On the optimality of the decision. All the arguments used for distinguishing both hypotheses are very crude and this raises the question whether a better test exists. It turns out that in the regime of interest to us, namely t and w linear in n, the term $\tilde{O}\left(\frac{1}{(\varepsilon_1-\varepsilon_0)^2}\right)$ is of the right order. Indeed our statistical test amounts actually to the Neymann-Pearson test (with a threshold in this case which is not necessarily in the middle, i.e. equal to $m\frac{1+\varepsilon_0+\varepsilon_1}{2}$). In the case of interest to us, the bias between both distributions $\varepsilon_1 - \varepsilon_0$ is exponentially small in n and Chernoff's bound captures accurately the large deviations of the random variable V_m. Now we could wonder whether using some finer knowledge about the hypotheses $\mathscr{H}_0$ and $\mathscr{H}_1$ could do better. For instance we know the a priori probabilities of these hypotheses since $\mathbb{P}(e_i = 1) = \frac{t}{n}$. It can be readily verified that using Bayesian hypothesis testing based on the a priori knowledge of the a priori probabilities of both hypotheses does not allow to change the order of number of tests which is still $\tilde{O}\left(\frac{1}{(\varepsilon_1-\varepsilon_0)^2}\right)$ when t and w are linear in n.

3.2 The statistical decoding algorithm

Statistical decoding is a randomized algorithm which uses the previous distinguisher. As we just noted, this distinguisher needs $\tilde{O}\left(\frac{1}{(\varepsilon_1-\varepsilon_0)^2}\right)$ parity-check equations of weight w to work. This number obviously depends on w, R and t and we use the notation:

Notation 3. $P_w \stackrel{\triangle}{=} \frac{1}{(\varepsilon_1-\varepsilon_0)^2}$.

Now we have two frameworks to present statistical decoding. We can consider the computation of $\tilde{O}(P_w)$ parity-check equations as a pre-computation or to consider it as a part of the algorithm. To consider the case of pre-computation, simply remove Line 4 of Algorithm 1 and consider the $\mathcal{S}_i$'s as an additional input to the algorithm. $\texttt{ParityCheckComputation}_w$ will denote an algorithm which for an input G, i outputs $\tilde{O}(P_w)$ vectors of $\mathcal{H}_{w,i}$.

Clearly statistical decoding complexity is given by

- When the $\mathcal{S}_i$'s are already stored and computed: $\tilde{O}\left(P_w\right)$;
- When the $\mathcal{S}_i$'s have to be computed: $\tilde{O}\Big(P_w + |\mathcal{PC}_w|\Big)$ where $|\mathcal{PC}_w|$ stands for the complexity of the call $\texttt{ParityCheckComputation}_w$.

As explained in introduction, our goal is to give the asymptotic complexity of statistical decoding. We introduce for this purpose the following notations:

Notation 4.
- $\omega \stackrel{\triangle}{=} \frac{w}{n}$;
- $\tau \stackrel{\triangle}{=} \frac{t}{n}$.

The two following quantities will be the central object of our study.

Algorithm 1 `DecoStat` : **Statistical Decoding**

Input : $G \in \mathbb{F}_2^{Rn \times n}, y = xG + e \in \mathbb{F}_2^n, w \in \mathbb{N}$
Output : e /**Error Vector**/
for $i = 1 \cdots n$ **do**
  $\mathcal{S}_i \leftarrow \texttt{ParityCheckComputation}_w(G, i)$ /**Auxiliary Algorithm**/
  $V_i \leftarrow 0$
  for all $h \in \mathcal{S}_i$ **do**
    $V_i \leftarrow V_i + \mathrm{sgn}(\varepsilon_1 - \varepsilon_0) \cdot \langle y, h \rangle$
  end for
  if $V_i < \mathrm{sgn}(\varepsilon_1 - \varepsilon_0) P_w \frac{1+\varepsilon_1+\varepsilon_0}{2}$ **then**
    $e_i \leftarrow 0$
  else
    $e_i \leftarrow 1$
  end if
end for
return e

Definition 1 (Asymptotic complexity of statistical decoding)**.** We define the asymptotic complexity of statistical decoding when the $\mathcal{S}_i$'s are already computed by

$$\pi(\omega, \tau) \stackrel{\triangle}{=} \underline{\lim}_{n \to +\infty} \frac{1}{n} \log_2 P_w$$

whereas the asymptotic complexity of the complete algorithm of statistical decoding (including the computation of the parity-check equations) is defined by

$$\pi^{complete}(\omega, \tau) \stackrel{\triangle}{=} \underline{\lim}_{n \to +\infty} \frac{1}{n} \max \Big(\log_2 P_w, \log_2 |\texttt{ParityCheckComputation}_w| \Big).$$

Remark 1. One could wonder why these quantities are defined as infimum limits and not directly as limits. This is due to the fact that in certain regions of the error weight and parity-check weights the asymptotic bias may from time to time become much smaller than it typically is. This bias is indeed proportional to values taken by a Krawtchouk polynomial and for certain errors weights and parity-check weights we may be close to the zero of the relevant Krawtchouk polynomial (this corresponds to the second case of Theorem 1).

We are looking for explicit formulas for $\pi(\omega, \tau)$ and $\pi^{complete}(\omega, \tau)$. The second quantity depends on the algorithm which is used. We will come back to this issue in Subsection 7.1. For our purpose we will use Krawtchouk polynomials and asymptotic expansions for them coming from [IS98]. Let m be a positive integer, we recall that the Krawtchouk polynomial of degree v and order m, $p_v^m(X)$ is defined for $v \in \{0, \cdots, m\}$ by:

$$p_v^m(X) = \frac{(-1)^v}{2^v} \sum_{j=0}^{v} (-1)^j \binom{X}{j} \binom{m-X}{v-j} \quad \text{where, } \binom{X}{j} = \frac{1}{j!} \left(X(X-1)\cdots(X-j+1)\right)$$

These Krawtchouk polynomials are readily related to our biases. We can namely observe that $\sum_{j=0}^{w-1} \binom{t-1}{j} \binom{n-t}{w-1-j} = \binom{n-1}{w-1}$ to recast the following evaluation of a Krawtchouk polyno-

mial as

$$\begin{aligned}
-\frac{(-2)^{w-2}}{\binom{n-1}{w-1}}p_{w-1}^{n-1}(t-1) &= \frac{\sum_{j=0}^{w-1}(-1)^j\binom{t-1}{j}\binom{n-t}{w-1-j}}{2\binom{n-1}{w-1}} \\
&= \frac{\sum_{\substack{j=0\\ j\text{ even}}}^{w-1}\binom{t-1}{j}\binom{n-t}{w-1-j} - \sum_{\substack{j=1\\ j\text{ odd}}}^{w-1}\binom{t-1}{j}\binom{n-t}{w-1-j}}{2\binom{n-1}{w-1}} \\
&= \frac{2\sum_{\substack{j=0\\ j\text{ even}}}^{w-1}\binom{t-1}{j}\binom{n-t}{w-1-j} - \binom{n-1}{w-1}}{2\binom{n-1}{w-1}} \\
&= \varepsilon_1 \qquad (1)
\end{aligned}$$

We have a similar computation for ε_0

$$\begin{aligned}
\frac{(-2)^{w-2}}{\binom{n-1}{w-1}}p_{w-1}^{n-1}(t) &= -\frac{\sum_{j=0}^{w-1}(-1)^j\binom{t}{j}\binom{n-1-t}{w-1-j}}{2\binom{n-1}{w-1}} \\
&= -\frac{\sum_{\substack{j=0\\ j\text{ even}}}^{w-1}\binom{t}{j}\binom{n-1-t}{w-1-j} - \sum_{\substack{j=1\\ j\text{ odd}}}^{w-1}\binom{t}{j}\binom{n-1-t}{w-1-j}}{2\binom{n-1}{w-1}} \\
&= -\frac{\binom{n-1}{w-1} - 2\sum_{\substack{j=0\\ j\text{ odd}}}^{w-1}\binom{t}{j}\binom{n-1-t}{w-1-j}}{2\binom{n-1}{w-1}} \\
&= \varepsilon_0 \qquad (2)
\end{aligned}$$

Let us recall Theorem 3.1 in [IS98].

Theorem 1 ([IS98, Th. 3.1])**.** *Let m, v and s be three positive integers. We set $\nu \stackrel{\triangle}{=} \frac{v}{m}, \alpha \stackrel{\triangle}{=} \frac{1}{\nu}$ and $\sigma = \frac{s}{m}$. We assume $\alpha \geq 2$. Let*

$$p(z) = \log_2 z - \frac{\sigma}{\nu}\log(1+z) - \left(\alpha - \frac{\sigma}{\nu}\right)\log_2(1-z).$$

$p'(z) = 0$ has two solutions x_1 and x_2 which are the two roots of the equation $(\alpha-1)X^2 + (\alpha - 2\frac{\sigma}{\nu})X + 1 = 0$. Let $D \stackrel{\triangle}{=} \left(\alpha - 2\frac{\sigma}{\nu}\right)^2 - 4(\alpha-1)$ and $\Delta \stackrel{\triangle}{=} \alpha - \frac{2\sigma}{\nu}$. The two roots are equal to $\frac{-\Delta \pm \sqrt{D}}{2(\alpha-1)}$ and x_1 is defined to be root $\frac{-\Delta+\sqrt{D}}{2(\alpha-1)}$. There are two cases to consider

- *In the case $\frac{\sigma}{\nu} \in (0, \alpha/2 - \sqrt{\alpha-1})$, D is positive, x_1 is a real negative number and we can write*

$$p_v^m(s) = Q_{\sigma,\nu}(v)2^{-(p(x_1)+1)v} \qquad (3)$$

  where $Q_{\sigma,\nu}(v) \stackrel{\triangle}{=} -\sqrt{\frac{1-r^2}{2\pi r D v}}(1 + O(v^{-1/2}))$ and $r \stackrel{\triangle}{=} -x_1$.

- *In the case $\frac{\sigma}{\nu} \in (\alpha/2 - \sqrt{\alpha-1}, \alpha/2)$, D is negative, x_1 is a complex number and we have*

$$p_v^m(s) = R_{\sigma,\nu}(v)\Im\left(\frac{2^{-(p(x_1)+1)v}}{x_1\sqrt{2p"(x_1)}}(1+\delta(v))\right) \qquad (4)$$

  where $\Im(z)$ denotes the imaginary part of the complex number z, $\delta(v)$ denotes a function which is $o(1)$ uniformly in v, and $R_{\sigma,\nu}(v) \stackrel{\triangle}{=} \frac{1+O(v^{-1/2})}{\sqrt{\pi v}}$.

The asymptotic formulas hold uniformly on the compact subsets of the corresponding open intervals.

Remark 1. Note that strictly speaking (3) is incorrectly stated in [IS98, Th. 3.1]. The problem is that (3.20) is incorrect in [IS98], since both $p"(-r_1)$ and $p^{(3)}(-r_1)$ are negative and taking a square root of these expressions leads to a purely imaginary number in (3.20). This can be easily fixed since the expression which is just above (3.20) is correct and it just remains to take the imaginary part correctly to derive (3).

It will be helpful to use the following notation from now on.

Notation 5.

$$\begin{aligned} m &\stackrel{\triangle}{=} n-1 \\ v &\stackrel{\triangle}{=} w-1 \\ \nu &\stackrel{\triangle}{=} \frac{v}{m} \\ \alpha &\stackrel{\triangle}{=} \frac{1}{\nu} \\ \sigma_0 &\stackrel{\triangle}{=} \frac{t}{m} \\ \sigma_1 &\stackrel{\triangle}{=} \frac{t-1}{m} \end{aligned}$$

and for $i \in \{0,1\}$ we define the following quantities

$$\begin{aligned} p_i(z) &\stackrel{\triangle}{=} \log_2 z - \frac{\sigma_i}{\nu}\log(1+z) - \left(\alpha - \frac{\sigma_i}{\nu}\right)\log_2(1-z) \\ \Delta_i &\stackrel{\triangle}{=} \alpha - \frac{2\sigma_i}{\nu} \\ D_i &\stackrel{\triangle}{=} \left(\alpha - 2\frac{\sigma_i}{\nu}\right)^2 - 4(\alpha-1) \\ z_i &\stackrel{\triangle}{=} \frac{-\Delta_i + \sqrt{D_i}}{2(\alpha-1)} \end{aligned}$$

We are now going use these asymptotic expansions to derive explicit formulas for $\pi(\omega,\tau)$. We start with the following lemma.

Lemma 2. *With the hypothesis of Proposition just above, we have*

$$\frac{\varepsilon_0}{\varepsilon_1} = -\frac{1+z_1}{1-z_1}\left(1+O(w^{-1/2}\right).$$

Proof. From (1) and (2) we have

$$\frac{\varepsilon_0}{\varepsilon_1} = -\frac{p_{w-1}^{n-1}(t)}{p_{w-1}^{n-1}(t-1)} \tag{5}$$

By using Theorem 1 we obtain when plugging the asymptotic expansions of the Krawtchouk polynomials into (5)

$$\begin{aligned} \frac{\varepsilon_0}{\varepsilon_1} &= -\frac{Q_{\sigma_0,\nu}(v)2^{-p_0(z_0)v}}{Q_{\sigma_1,\nu}(v)2^{-p_1(z_1)v}} \\ &= -\frac{Q_{\sigma_0,\nu}(v)}{Q_{\sigma_1,\nu}(v)}2^{(p_1(z_1)-p_0(z_0))v} \end{aligned} \tag{6}$$

We clearly have $\sigma_1 = \sigma_0 - \frac{1}{m}$ and $z_1 = z_0 + O\left(\frac{1}{m}\right)$ and therefore from the particular form of $Q_{\sigma_i,\nu}(v)$ we deduce that

$$\frac{Q_{\sigma_0,\nu}(v)}{Q_{\sigma_1,\nu}(v)} = 1 + O(v^{-1/2}). \tag{7}$$

We observe now that

$$\begin{aligned} \frac{\sigma_1}{\nu} - \frac{\sigma_0}{\nu} &= \frac{t-1}{v} - \frac{t}{v} && (8)\\ &= -\frac{1}{v} && (9) \end{aligned}$$

and therefore

$$\begin{aligned} (p_1(z_1) - p_0(z_0))v = &\Big(\log_2(z_1) - \frac{\sigma_1}{\nu}\log_2(1+z_1) - (\alpha - \frac{\sigma_1}{\nu})\log_2(1-z_1)\\ &- \log_2(z_0) + \frac{\sigma_0}{\nu}\log_2(1+z_0) + (\alpha - \frac{\sigma_0}{\nu})\log_2(1-z_0)\Big)\, v\\ = &\left(\log_2\frac{z_1}{z_0} - \frac{\sigma_0}{\nu}\log_2\frac{1+z_1}{1+z_0} - (\alpha - \frac{\sigma_0}{\nu})\log_2\frac{1-z_1}{1-z_0} + \frac{1}{v}\log_2(1+z_1) - \frac{1}{v}\log_2(1-z_1)\right) v\\ = &\left(\log_2\frac{z_1}{z_0} - \frac{\sigma_0}{\nu}\log_2\frac{1+z_1}{1+z_0} - (\alpha - \frac{\sigma_0}{\nu})\log_2\frac{1-z_1}{1-z_0}\right) v + \log_2\frac{1+z_1}{1-z_1} \qquad (10) \end{aligned}$$

It is insightful to express the term $\log_2\frac{z_1}{z_0} - \frac{\sigma_0}{\nu}\log_2\frac{1+z_1}{1+z_0} - (\alpha - \frac{\sigma_0}{\nu})\log_2\frac{1-z_1}{1-z_0}$ as

$$\log_2\frac{z_1}{z_0} - \frac{\sigma_0}{\nu}\log_2\frac{1+z_1}{1+z_0} - (\alpha - \frac{\sigma_0}{\nu})\log_2\frac{1-z_1}{1-z_0} \;=\; p_0(z_1) - p_0(z_0)$$

The point is that $p_0'(z_0) = 0$ and $z_1 = z_0 + \delta$ where $\delta = O(1/m)$. Therefore

$$p_0(z_1) - p_0(z_0) = p_0(z_0+\delta) - p_0(z_0) = O(\delta^2) = O(1/m^2).$$

Using this in (10) and then in (6) implies the lemma. □

From this lemma we can deduce that

Lemma 3. *Assume $\alpha \geq 2$ and $\frac{\sigma_i}{\nu} \in (0, \alpha/2 - \sqrt{\alpha-1})$ for $i \in \{0,1\}$. We have*

$$\varepsilon_0 - \varepsilon_1 = (-1)^v \sqrt{\frac{(1+z_1)(1-\nu)}{(z_1 - z_1^2)D_1}}\, 2^{-\left(p(z_1) + \frac{H(\nu)}{\nu}\right)v}(1 + O(w^{-1/2})).$$

Proof. We have

$$\begin{aligned}
\varepsilon_0 - \varepsilon_1 &= -\varepsilon_1 \frac{1+z_1}{1-z_1}(1+O(w^{-1/2})) - \varepsilon_1 \\
&= -\varepsilon_1 \left(\frac{1+z_1}{1-z_1}(1+O(w^{-1/2}) + 1 \right) \\
&= -2\varepsilon_1 \left(\frac{1}{1-z_1} + O\left(w^{-1/2} \frac{1+z_1}{1-z_1} \right) \right) \\
&= -\frac{2\varepsilon_1}{1-z_1}(1+O(w^{-1/2}) \quad (\text{since } -1 \le z_i \le 0) \\
&= -\frac{(-2)^{v-1}\sqrt{2\pi v(1-\nu)}}{2^{\frac{vH(\nu)}{\nu}}} Q_{\sigma_1,\nu}(v) 2^{-(p(z_1)+1)v} \frac{2}{1-z_1}(1+O(w^{-1/2}) \qquad (11) \\
&= \frac{(-2)^v \sqrt{2\pi v(1-\nu)}}{2^{\frac{vH(\nu)}{\nu}}} \sqrt{\frac{1-z_1^2}{-2\pi z_1 D_1 v}} 2^{-(p(z_1)+1)v} \frac{2}{1-z_1}(1+O(w^{-1/2}) \\
&= (-1)^v \sqrt{\frac{(1+z_1)(1-\nu)}{(z_1-z_1^2)D_1}} 2^{-\left(p(z_1)+\frac{H(\nu)}{\nu}\right)v}(1+O(w^{-1/2}) \qquad (12)
\end{aligned}$$

where we used in (11)

$$\begin{aligned}
\binom{m}{v} &= \frac{2^{mH(\nu)}}{\sqrt{2\pi m\nu(1-\nu)}} \\
&= \frac{2^{\frac{vH(\nu)}{\nu}}}{\sqrt{2\pi v(1-\nu)}}
\end{aligned}$$

□

The second case corresponding to $\frac{\sigma_i}{\omega} \in (\alpha/2 - \sqrt{\alpha-1}, \alpha/2)$ is handled by the following lemma (note that it is precisely the "sin" term that appears in it that lead us to define $\pi(\omega,\tau)$ as an infimum limit and not as a limit)

Lemma 4. *When $\frac{\sigma_i}{\omega} \in (\alpha/2 - \sqrt{\alpha-1}, \alpha/2)$ for $i \in \{0,1\}$ we have*

$$\varepsilon_1 - \varepsilon_0 = \frac{(-1)^v \sqrt{1-\nu}}{\left|(z_0 - z_0^2)\sqrt{p_0"(z_0)}\right|} 2^{-v\left(\Re(p_0(z_0)) + \frac{H(\nu)}{\nu}\right)} \sin\left(v\theta - \theta_0 + o(1)\right)(1+o(1))$$

where $\theta \stackrel{\triangle}{=} \arg\left(2^{-p_0(z_0)}\right)$ and $\theta_0 \stackrel{\triangle}{=} \arg\left((z_0 - z_0^2)\sqrt{p_0"(z_0)}\right)$.

Proof. The proof of this lemma is very similar to the proof of Lemma 2. From (1) and (2) we have

$$\varepsilon_1 - \varepsilon_0 = -\frac{(-2)^{w-2}}{\binom{n-1}{w-1}}\left(p_{w-1}^{n-1}(t) + p_{w-1}^{n-1}(t-1)\right) \qquad (13)$$

By plugging the asymptotic expansion of Krawtchouk polynomials given in Theorem 1 into (13) we obtain

$$\varepsilon_1 - \varepsilon_0 = -\frac{(-2)^{w-2}}{\binom{n-1}{w-1}} \left(R_{\sigma_1,\nu}(v) \Im\left(\frac{2^{-(p_1(z_1)+1)v}}{z_1\sqrt{2p_1"(z_1)}}(1+\delta_1(v)) \right) + R_{\sigma_0,\nu}(v) \Im\left(\frac{2^{-(p_0(z_0)+1)v}}{z_0\sqrt{2p_0"(z_0)}}(1+\delta_0(v)) \right) \right)$$

where the δ_i's are functions which are of order $o(1)$ uniformly in v.

We clearly have $\sigma_1 = \sigma_0 - \frac{1}{m}$ and $z_1 = z_0 + O\left(\frac{1}{m}\right)$ and therefore from the particular form of $R_{\sigma_i,\nu}(v)$ we deduce that

$$\begin{aligned} R_{\sigma_1,\nu}(v) &= R_{\sigma_0,\nu}(v)\left(1+O(v^{-1/2})\right) \\ \frac{1}{z_1\sqrt{2p_1"(z_1)}} &= \frac{1}{z_0\sqrt{2p_0"(z_0)}}\left(1+O\left(\frac{1}{m}\right)\right) \end{aligned}$$

From this we deduce that

$$\begin{aligned} \varepsilon_1-\varepsilon_0 &= \frac{(-1)^v}{2\binom{n-1}{w-1}}R_{\sigma_0,\nu}(v)\left(\Im\left(\frac{2^{-p_1(z_1)v}}{z_0\sqrt{2p_0"(z_0)}}(1+o(1))\right)+\Im\left(\frac{2^{-p_0(z_0)v}}{z_0\sqrt{2p_0"(z_0)}}(1+\delta_0(v))\right)\right) \\ &= \frac{(-1)^v}{2\binom{n-1}{w-1}}R_{\sigma_0,\nu}(v)\Im\left(\left(\frac{2^{-p_0(z_0)v}}{z_0\sqrt{2p_0"(z_0)}}\left(1+\delta_0(v)+2^{(p_0(z_0)-p_1(z_1))v}(1+o(1))\right)\right)\right) \quad (14) \end{aligned}$$

We now observe that

$$\begin{aligned} p_0(z_0)-p_1(z_1) &= \log_2(z_0)-\frac{\sigma_0}{\nu}\log_2(1+z_0)-(\alpha-\frac{\sigma_0}{\nu})\log_2(1-z_0) \quad (15) \\ &\quad -\log_2(z_1)+\frac{\sigma_1}{\nu}\log_2(1+z_1)+(\alpha-\frac{\sigma_1}{\nu})\log_2(1-z_1) \\ &= \log_2\frac{z_0}{z_1}-\frac{\sigma_0}{\nu}\log_2\frac{1+z_0}{1+z_1}-(\alpha-\frac{\sigma_0}{\nu})\log_2\frac{1-z_0}{1-z_1}+\left(\frac{\sigma_1}{\nu}-\frac{\sigma_0}{\nu}\right)\log_2\frac{1+z_1}{1-z_1} \\ &= \log_2\frac{z_0}{z_1}-\frac{\sigma_0}{\nu}\log_2\frac{1+z_0}{1+z_1}-(\alpha-\frac{\sigma_0}{\nu})\log_2\frac{1-z_0}{1-z_1}-\frac{1}{v}\log_2\frac{1+z_1}{1-z_1} \quad (16) \end{aligned}$$

where (16) follows from the observation

$$\begin{aligned} \frac{\sigma_1}{\nu}-\frac{\sigma_0}{\nu} &= \frac{t-1}{v}-\frac{t}{v} \\ &= -\frac{1}{v} \end{aligned}$$

Recall that $z_1 = z_0+\delta$ where $\delta = O(1/m)$ and that

$$\begin{aligned} \log_2\frac{z_0}{z_1}-\frac{\sigma_0}{\nu}\log_2\frac{1+z_0}{1+z_1}-(\alpha-\frac{\sigma_0}{\nu})\log_2\frac{1-z_0}{1-z_1} &= p_0(z_0)-p_0(z_1) \\ &= p_0(z_0)-p_0(z_0+\delta) \end{aligned}$$

The point is that $p_0'(z_0)=0$ and therefore

$$p_0(z_0)-p_0(z_0+\delta) = O(\delta^2) = O(1/m^2).$$

Using this in (16) and then multiply by v implies

$$(p_0(z_0)-p_1(z_1))v = -\log_2\frac{1-z_1}{1+z_1}+O(1/v) = -\log_2\frac{1-z_0}{1+z_0}+O(1/v) \quad (17)$$

We can substitute for this expression in (14) and obtain

$$\begin{aligned} \varepsilon_1-\varepsilon_0 &= \frac{(-1)^v}{2\binom{n-1}{w-1}}R_{\sigma_0,\nu}(v)\Im\left(\left(\frac{2^{-p_0(z_0)v}}{z_0\sqrt{2p_0"(z_0)}}\left(1+\frac{1+z_0}{1-z_0}+o(1)\right)\right)\right) \\ &= \frac{(-1)^v}{\binom{m}{v}}R_{\sigma_0,\nu}(v)\Im\left(\left(\frac{2^{-p_0(z_0)v}}{(z_0-z_0^2)\sqrt{2p_0"(z_0)}}(1+o(1))\right)\right) \quad (18) \end{aligned}$$

Recall that

$$\begin{aligned} R_{\sigma,\nu}(v) &= \frac{1+o(1)}{\sqrt{\pi v}} \\ \binom{m}{v} &= \frac{2^{mH(\nu)}}{\sqrt{2\pi m\nu(1-\nu)}} \\ &= \frac{2^{\frac{vH(\nu)}{\nu}}}{\sqrt{2\pi v(1-\nu)}} \end{aligned}$$

By using this in (18) we obtain

$$\varepsilon_1 - \varepsilon_0 = \frac{(-1)^v\sqrt{2\pi v(1-\nu)}}{\sqrt{\pi v}\left|(z_0 - z_0^2)\sqrt{2p_0"(z_0)}\right|} 2^{-v\left(\Re(p_0(z_0))+\frac{H(\nu)}{\nu}\right)} \sin\left(v\theta - \theta_0\right)(1+o(1)) \quad (19)$$

$$= \frac{(-1)^v\sqrt{1-\nu}}{\left|(z_0 - z_0^2)\sqrt{p_0"(z_0)}\right|} 2^{-v\left(\Re(p_0(z_0))+\frac{H(\nu)}{\nu}\right)} \sin\left(v\theta - \theta_0 + o(1)\right)(1+o(1)) \quad (20)$$

□

From Lemmas 3 and 4 we deduce immediately that

Corollary 5. *We set* $\gamma = \frac{1}{\omega}$,

- *If* $\frac{\tau}{\omega} \in (0, \gamma/2 - \sqrt{\gamma-1})$*:*

$$\pi(\omega,\tau) = 2\left(\omega\left(\log_2(r) - \frac{\tau}{\omega}\log_2(1-r) - (\gamma - \frac{\tau}{\omega})\log_2(1+r)\right) + H(\omega)\right)$$

where r *is the smallest root of* $(\gamma-1)X^2 - (\gamma - 2\frac{\tau}{\omega})X + 1$

- *If* $\frac{\tau}{\omega} \in (\gamma/2 - \sqrt{\gamma-1}, \gamma/2)$*:*

$$\pi(\omega,\tau) = 2\left(\omega\Re\left(\log_2(z) - \frac{\tau}{\omega}\log_2(1+z) - (\gamma - \frac{\tau}{\omega})\log_2(1-z)\right) + H\left(\omega\right)\right)$$

where $z = re^{\mathbf{i}\varphi}$ *with* $r = \frac{1}{\sqrt{\gamma-1}}$ *and* $\cos(\varphi) = \frac{2\frac{\tau}{\omega} - \gamma}{2\sqrt{\gamma-1}}$

Remark 2. These asymptotic formulas turn out to be already accurate in the "cryptographic range" as it is shown in Figure 1.

Amazingly enough these formulas can be simplified a lot in the second case of the corollary as shown by the following theorem.

Theorem 2 (Asymptotic complexity of statistical decoding)**.**

- *If* $\tau \in \left(0, \frac{1}{2} - \sqrt{\omega - \omega^2}\right)$*:* $\pi(\omega,\tau) = 2\omega\log_2(r) - 2\tau\log_2(1-r) - 2(1-\tau)\log_2(1+r) + 2H(\omega)$ *where* r *is the smallest root of* $(1-\omega)X^2 - (1-2\tau)X + \omega = 0$.
- *If* $\tau \in \left(\frac{1}{2} - \sqrt{\omega - \omega^2}, \frac{1}{2}\right)$*:* $\pi(\omega,\tau) = H(\omega) + H(\tau) - 1$.

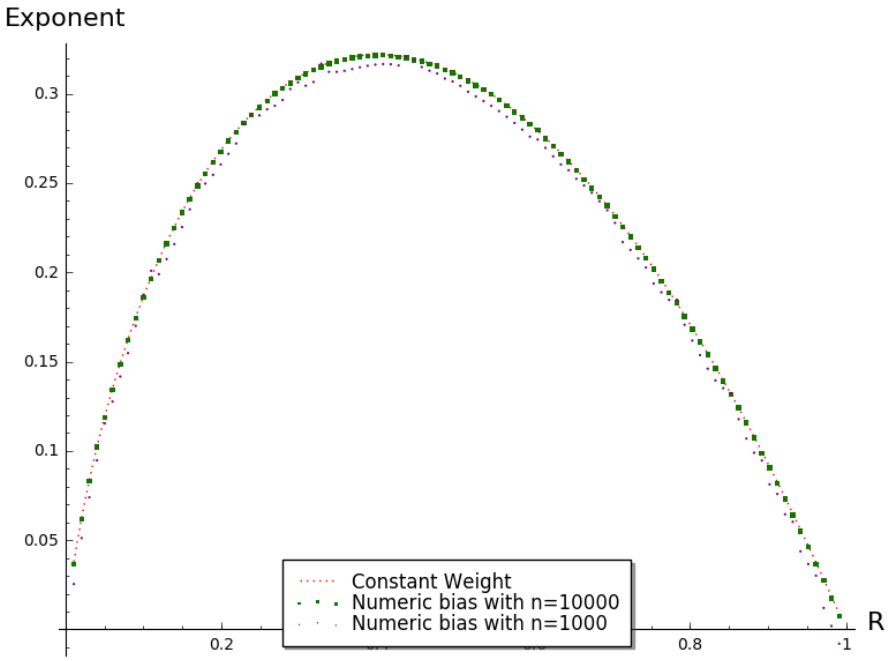

Figure 1: Comparison of the asymptotic and numeric exponents for $\tau = H^{-1}(1-R)$.

Proof. The first case is just a slight rewriting. To prove the formula corresponding to the second case let us recall that the z that appears in the second case of Corollary 5 satisfies $p'(z) = 0$ where

$$p(z) \stackrel{\triangle}{=} \omega \log_2 z - \tau \log_2(1+z) - (1-\tau)\log_2(1-z).$$

Let

$$\begin{aligned} f(\omega,\tau) &\stackrel{\triangle}{=} 2\left(\omega\Re\left(\log_2(z) - \frac{\tau}{\omega}\log_2(1+z) - (\gamma - \frac{\tau}{\omega})\log_2(1-z)\right) + H(\omega)\right) \\ &= 2\Re(p(z)) + 2H(\omega). \end{aligned}$$

Let us first differentiate this expression with respect to ω:

$$\begin{aligned} \frac{\partial f(\omega,\tau)}{\partial \omega} &= 2\Re(p'(z))\frac{\partial z}{\partial \omega} + 2\Re(\log_2(z)) + 2\log_2\frac{1-\omega}{\omega} \\ &= 2\Re(\log_2(z)) + 2\log_2\left(\frac{1-\omega}{\omega}\right) \end{aligned} \tag{21}$$

Since $z = re^{\mathbf{i}\varphi}$ with $r = \frac{1}{\sqrt{\gamma-1}}$, we deduce that

$$2\Re(\log_2(z)) = 2\log_2 r = 2\log_2\left(\frac{1}{\sqrt{\gamma-1}}\right) = \log_2\left(\frac{1}{1/\omega - 1}\right) = \log_2\left(\frac{\omega}{1-\omega}\right).$$

Substituting this expression for $2\Re(\log_2(z))$ in (21) yields

$$\frac{\partial f(\omega,\tau)}{\partial \omega} = \log_2\left(\frac{\omega}{1-\omega}\right) + 2\log_2\left(\frac{1-\omega}{\omega}\right) = \log_2\left(\frac{1-\omega}{\omega}\right) = H'(\omega). \tag{22}$$

We continue the proof by differentiating now $f(\omega,\tau)$ with respect to τ:

$$\begin{aligned}\frac{\partial f(\omega,\tau)}{\partial\tau} &= 2\Re(p'(z))\frac{\partial z}{\partial\tau} - 2\Re\left(\log_2(1+z) - \log_2(1-z)\right)\\ &= -2\Re\left(\log_2\left(\frac{1+z}{1-z}\right)\right)\end{aligned}$$

Recall that z is also given by one of the two roots of $(1-\omega)X^2+(1-2\tau)X+\omega=0$ (see Theorem 1 for the root which is actually chosen) and therefore

$$z = \frac{2\tau - 1 + \mathbf{i}\sqrt{4\omega(1-\omega)-(1-2\tau)^2}}{2(1-\omega)}$$

From this we deduce that

$$\begin{aligned}1+z &= \frac{1-2\omega+2\tau+\mathbf{i}\sqrt{4\omega(1-\omega)-(1-2\tau)^2}}{2(1-\omega)}\\ 1-z &= \frac{3-2\omega-2\tau-\mathbf{i}\sqrt{4\omega(1-\omega)-(1-2\tau)^2}}{2(1-\omega)}\end{aligned}$$

$$\begin{aligned}-2\Re\left(\log_2\left(\frac{1+z}{1-z}\right)\right) &= -2\Re\left(\log_2\left(\frac{1-2\omega+2\tau+\mathbf{i}\sqrt{4\omega(1-\omega)-(1-2\tau)^2}}{3-2\omega-2\tau-\mathbf{i}\sqrt{4\omega(1-\omega)-(1-2\tau)^2}}\right)\right)\\ &= -2\log_2\left|\frac{1-2\omega+2\tau+\mathbf{i}\sqrt{4\omega(1-\omega)-(1-2\tau)^2}}{3-2\omega-2\tau-\mathbf{i}\sqrt{4\omega(1-\omega)-(1-2\tau)^2}}\right|\\ &= -\log_2\frac{(1-2\omega+2\tau)^2+4\omega(1-\omega)-(1-2\tau)^2}{(3-2\omega-2\tau)^2+4\omega(1-\omega)-(1-2\tau)^2}\\ &= -\log_2\frac{1+4\omega^2+4\tau^2-4\omega+4\tau-8\omega\tau+4\omega-4\omega^2-1-4\tau^2+4\tau}{9+4\omega^2+4\tau^2-12\omega-12\tau+8\omega\tau+4\omega-4\omega^2-1-4\tau^2+4\tau}\\ &= -\log_2\frac{8\tau-8\omega\tau}{8-8\omega-8\tau+8\omega\tau}\\ &= -\log_2\frac{8\tau(1-\omega)}{8(1-\omega)(1-\tau)}\\ &= -\log_2\frac{\tau}{1-\tau}\\ &= H'(\tau)\end{aligned}$$

These two results on the derivative imply that

$$f(\omega,\tau) = H(\omega) + H(\tau) + C$$

for some constant C which is easily seen to be equal to -1 by letting ω go to 0 and τ go to $\frac{1}{2}$ in $f(\omega,\tau)$.

□

4 The binomial model

[FKI07] introduced another model for the parity-check equations used in statistical decoding. Instead of assuming that they are chosen randomly of a given weight w, the authors of [FKI07] assume that they are random binary words of length n where the entries are chosen independently of each other according to a Bernoulli distribution of parameter w/n. In other words, the expected weight is still w but the weight of the parity-check equation is not fixed anymore and may vary. We will call it the *binomial model* of weight w and length n and refer to our model as the constant weight model of weight w. The binomial model presents the advantage of simplifying significantly the analysis of statistical decoding. It is easy to analyze the simple statistical decoding algorithm that we consider here and to compute asymptotically the number of parity-check equations that ensure successful decoding. We will do this in what follows. But the authors of [FKI07] went further since they were even able to analyze asymptotically an iterative version of statistical decoding by following some of the ideas of [SV04]. They showed that

Proposition 6 ([FKI07, Proposition 2.1 p.405])**.** *In the binomial model of weight w and length n, the number of check sums that are necessary to correct with large enough probability t errors by using the iterative decoding algorithm of [FKI07] is well estimated by $O(J_{min})$ with*

$$J_{min} = \left(\frac{n}{n-2w}\right)^{2(t-1)} = \left(1-\frac{2w}{n}\right)^{-2(t-1)}$$

where the constant in the "big O" depends on the ratio t/n.

Let us first show that naive statistical decoding performs almost as well when we forget about polynomial factors. It makes sense in order to compare both models to introduce some additional notation.

$$\begin{aligned} q_0^{\text{bin}} &= \mathbb{P}^{\text{bin}}\left(\langle e,h\rangle = 1 | h_i = 1\right) \text{ when } e_i = 0 \\ q_1^{\text{bin}} &= \mathbb{P}^{\text{bin}}\left(\langle e,h\rangle = 1 | h_i = 1\right) \text{ when } e_i = 1 \end{aligned}$$

where h is a parity-check equation chosen according to the binomial model and the probability is taken over the random choice of h in this model (and $\mathbb{P}^{\text{bin}}$ means that we take the probabilities according to the binomial model). These quantities do not depend on i. It will also be convenient to define $\varepsilon_0^{\text{bin}}$ and $\varepsilon_0^{\text{bin}}$ as

$$q_0^{\text{bin}} = \frac{1}{2} + \varepsilon_0^{\text{bin}} \;\; ; \;\; q_1^{\text{bin}} = \frac{1}{2} + \varepsilon_1^{\text{bin}}.$$

The computations of [FKI07, Sec II. B] show that

$$\begin{aligned} q_0^{\text{bin}} &= \frac{1-\left(1-\frac{2w}{n}\right)^t}{2} \\ q_1^{\text{bin}} &= \frac{1+\left(1-\frac{2w}{n}\right)^{t-1}}{2} \end{aligned}$$

This implies that

$$\varepsilon_0^{\text{bin}} = -\frac{\left(1-\frac{2w}{n}\right)^t}{2} \;\; ; \;\; \varepsilon_1^{\text{bin}} = \frac{\left(1-\frac{2w}{n}\right)^{t-1}}{2}.$$

It is also convenient in order to distinguish both models to rename the quantities q_0, q_1, ε_0 and ε_1 that were introduced before by referring to them as q_0^{con}, q_1^{con}, $\varepsilon_0^{\mathrm{con}}$ and $\varepsilon_1^{\mathrm{con}}$ respectively. We can perform the same statistical test as before by computing from m parity-check equations $h^1, \ldots, h^m$ all involving the bit i we want to decode, the quantity

$$V_m = \sum_{k=1}^{m} \mathrm{sgn}(\varepsilon_1^{\mathrm{bin}} - \varepsilon_0^{\mathrm{bin}}) \langle y, h^k \rangle = \sum_{k=1}^{m} \langle y, h^k \rangle.$$

The expectation of this quantity is $E_b \stackrel{\triangle}{=} m\left(\frac{1}{2} + \varepsilon_b^{\mathrm{bin}}\right)$ depending on the value $b \in \{0,1\}$ of the bit we want to decode. We decide that the bit we want to decode is equal to 0 if $V_m < \frac{E_0+E_1}{2}$ and 1 otherwise. As before, we observe that by Chernoff's bound we make a wrong decision with probability at most $2 \cdot 2^{-m\frac{(\varepsilon_1^{\mathrm{bin}} - \varepsilon_0^{\mathrm{bin}})^2}{2\ln(2)}}$. This probability can be made to be of order $o(1/n)$ by choosing m as $m = K \log n \frac{1}{(\varepsilon_1^{\mathrm{bin}} - \varepsilon_0^{\mathrm{bin}})^2}$ for a suitable constant K. In this case, decoding the whole sequence succeeds with probability $1 - o(1)$. In other words, naive statistical decoding succeeds for $m = O\left(\log n \frac{1}{(\varepsilon_1^{\mathrm{bin}} - \varepsilon_0^{\mathrm{bin}})^2}\right)$.

We may observe now that

$$\begin{aligned} \frac{1}{(\varepsilon_1^{\mathrm{bin}} - \varepsilon_0^{\mathrm{bin}})^2} &= O\left((1-2w/n)^{-2t-1}\right) \\ &= O(J_{\min}) \end{aligned}$$

This means that naive statistical decoding needs only marginally more equations in the binomial model (namely a multiplicative factor of order $O(\log n)$). To summarize the whole discussion, the number of parity-checks needed for decoding is

- with iterative statistical decoding over the binomial model

$$O\left(\frac{1}{(\varepsilon_1^{\mathrm{bin}} - \varepsilon_0^{\mathrm{bin}})^2}\right),$$

- with naive statistical decoding over the binomial model

$$O\left(\frac{\log n}{(\varepsilon_1^{\mathrm{bin}} - \varepsilon_0^{\mathrm{bin}})^2}\right)$$

- with naive statistical decoding over the constant weight model

$$O\left(\frac{\log n}{(\varepsilon_1^{\mathrm{con}} - \varepsilon_0^{\mathrm{con}})^2}\right).$$

One might wonder now whether there is a difference between both models. It is very tempting to conjecture that both models are very close to each other since the expected weight of the parity-checks is w in both cases. However this is not the case, we are really in a large deviation situation where the bias of some extreme weights take over the bias corresponding to the typical weight of the parity check equations. To illustrate this point, we choose the weight to be $w = \omega n$, the number of errors as $t = \tau n$ for some fixed ω and τ, and then let

n go to infinity. The normalized exponent[1] of the number of parity-check equations which is needed is

$$\lim_{n\to+\infty} \frac{1}{n} \log_2 \left(\frac{1}{(\varepsilon_1^{\text{bin}} - \varepsilon_0^{\text{bin}})^2} \right) = -2\tau \log_2 (1-2\omega)$$

in the binomial case, whereas $\lim_{n\to+\infty} \frac{1}{n} \log_2 \left(\frac{1}{(\varepsilon_1^{\text{con}} - \varepsilon_0^{\text{con}})^2} \right)$ is given by Theorem 2 in the constant weight case and both terms are indeed different in general. One case which is particularly interesting is when τ and ω are chosen as $\tau = H^{-1}(1-R)$ and $\omega = R/2$, where R is the code rate we consider. This corresponds to the hardest case of syndrome decoding and when the parity-check equations of this weight can be easily obtained as we will see in Section 6. The two normalized exponents are compared on Figure 2 as a function of the rate R. As we see, there is a huge difference. The problem with the model chosen in [FKI07] is that it is a very favorable model for statistical decoding. To the best of our knowledge there are no efficient algorithms for producing such parity-checks when $\omega \leq R/2$. Note that even such an algorithm were to exist, selecting appropriately only one weight would not change the exponential complexity of the algorithm (this will be proved in Section 5). In other words, in order to study statistical decoding we may restrict ourselves, as we do here, to considering only one weight and not a whole range of weights.

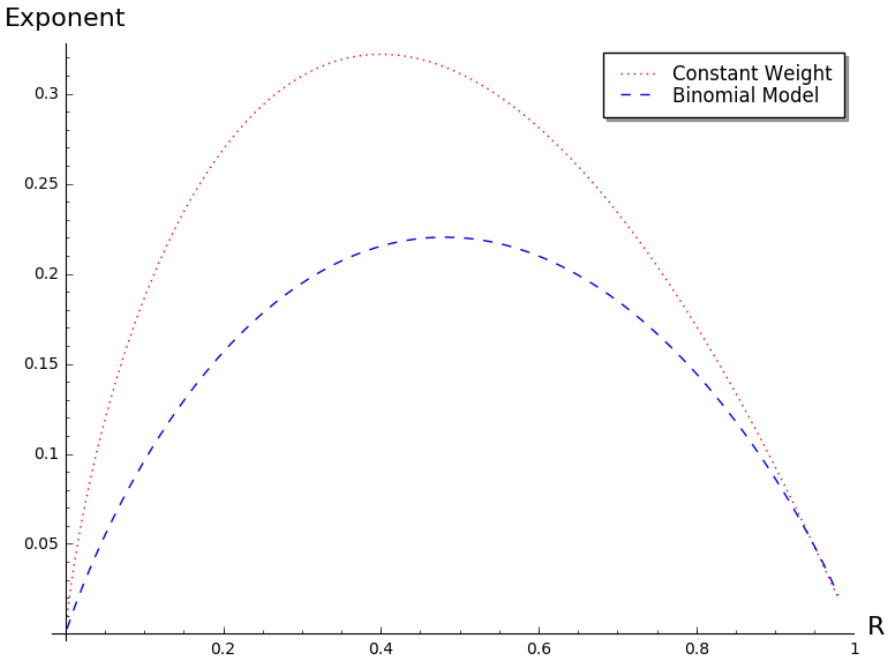

Figure 2: Comparison of the normalized exponents with $\tau = H^{-1}(1-R)$ of the number of parity-check equations which are needed in the binomial and the constant weight case.

The difference between both formulas is even more apparent when considering the slopes at the origin as shown in Figure 3. However both models get closer when the error weight decreases. For instance when considering a relative error $\tau = H^{-1}(1-R)/2$, we see in Figure

[1]Here the number of equations is a function of the form $\tilde{O}\left(e^{\alpha(\tau,\omega)n}\right)$ and we mean here the coefficient $\alpha(\omega,\tau)$.

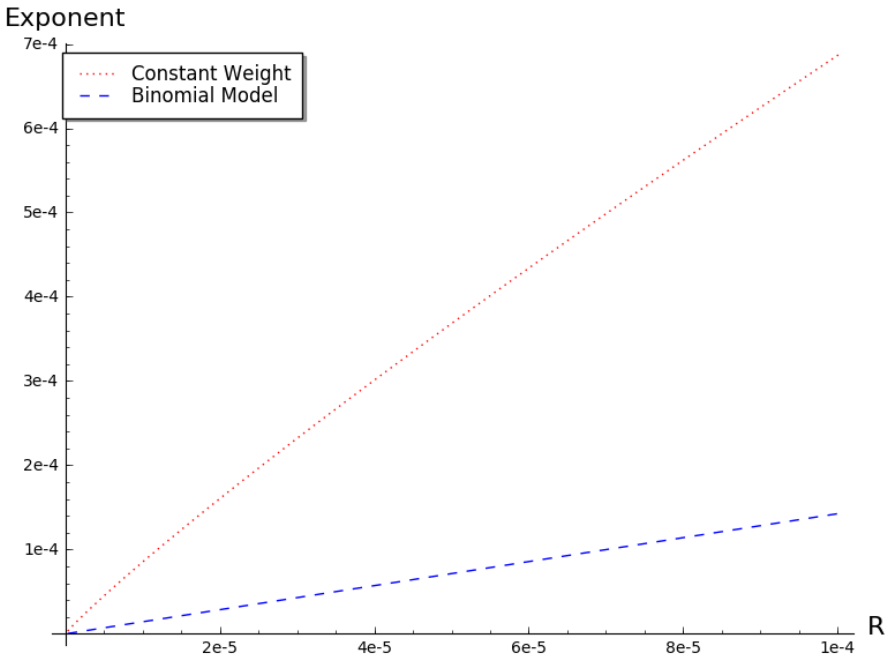

Figure 3: Comparison of the complexities with $\tau = H^{-1}(1-R)$ for rate close to 0

4 that the difference between both models gets significantly smaller. Actually the difference vanishes when the relative error tends to 0, as shown by Proposition 7.

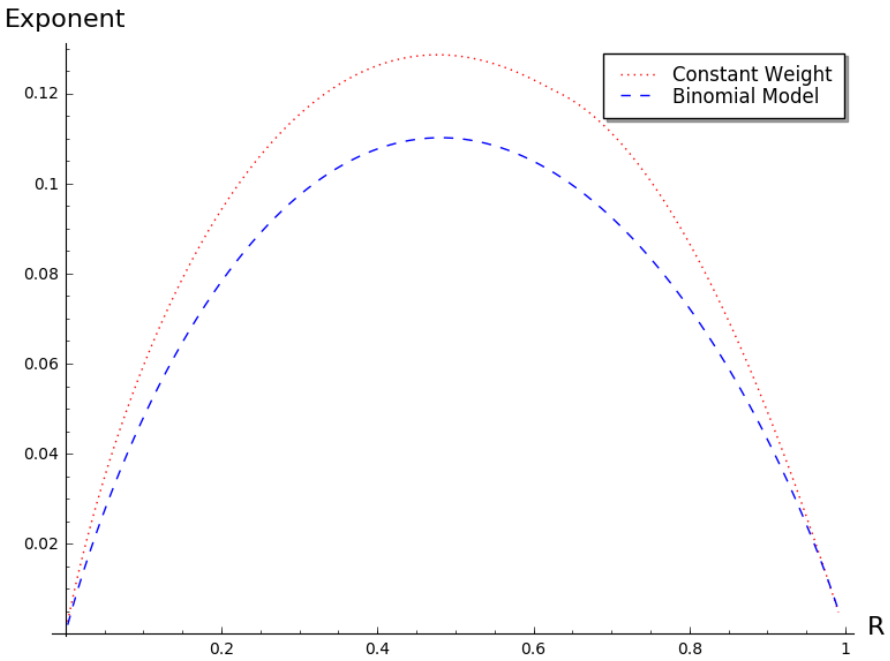

Figure 4: Comparison of the normalized exponents with $\tau = H^{-1}(1-R)/2$ of the number of parity-check equations which are needed in the binomial and the constant weight case.

Proposition 7 (Asymptotic complexity of statistical decoding for a sub-linear error weight)**.**

$$\pi(\omega,\tau) \underset{\tau\to 0}{=} -2\tau\log_2(1-2\omega)+o(\tau)$$

Proof. As τ decreases to 0, we consider for $\pi(\omega,\tau)$ the first formula which is given in Theorem 2. We have:

$$\begin{aligned}\pi(\omega,\tau) &= 2\omega\log_2(r)-2\tau\log_2(1-r)-2(1-\tau)\log_2(1+r)+2H(\omega)\\ &= 2\omega\log_2(r)-2\log_2(1+r)-2\tau\log_2\left(\frac{1-r}{1+r}\right)+2H(\omega)\end{aligned} \tag{23}$$

with

$$r=\frac{1-2\tau-\sqrt{(1-2\tau)^2-4\omega(1-\omega)}}{2(1-w)}$$

Let us compute now Taylor series expansion of r when $\tau\to 0$. We start with

$$\begin{aligned}r &= \frac{1-2\tau-\sqrt{1-4\omega(1-\omega)-4\tau+4\tau^2}}{2(1-\omega)}\\ &= \frac{1-2\tau-\sqrt{(1-2\omega)^2-4\tau+o(\tau)}}{2(1-\omega)}\end{aligned}$$

Now using the fact that:

$$(A^2-\varepsilon)^{1/2} \underset{\varepsilon\to 0}{=} A-\frac{\varepsilon}{2A}+o(\varepsilon)$$

we have:

$$\begin{aligned}r &= \frac{1-2\tau-(1-2\omega)+\frac{2\tau}{1-2\omega}+o(\tau)}{2(1-\omega)}\\ &= \frac{\omega}{1-\omega}+\frac{\tau\omega}{(1-\omega)(1-2\omega)}+o(\tau)\end{aligned}$$

And we deduce that:

$$1-r=\frac{1-2\omega}{1-\omega}-\frac{\tau\omega}{(1-\omega)(1-2\omega)}+o(\tau)$$

$$1+r=\frac{1}{1-\omega}+\frac{\tau\omega}{(1-\omega)(1-2\omega)}+o(\tau)$$

and therefore

$$-2\tau\log_2\left(\frac{1-r}{1+r}\right) \underset{\tau\to 0}{=} -2\tau\log_2(1-2\omega)+o(\tau) \tag{24}$$

Now using the fact that:

$$\log_2(A+\varepsilon) \underset{\varepsilon\to 0}{=} \log_2(A)+(1/\ln(2))\left(\frac{\varepsilon}{A}+o(\varepsilon)\right)$$

we have the asymptotic expansions with the logarithms:

$$\log_2(r)=\log_2\left(\frac{\omega}{1-\omega}\right)+(1/\ln(2))\left(\frac{\tau}{1-2\omega}+o(\tau)\right)$$

$$\log_2(1+r) = \log_2\left(\frac{1}{1-\omega}\right) + (1/\ln(2))\left(\frac{\tau\omega}{1-2\omega} + o(\tau)\right)$$

So we deduce that:

$$\begin{aligned} 2\omega\log_2(r) - 2\log_2(1+r) &= 2\omega\log_2\left(\frac{\omega}{1-\omega}\right) - 2\log_2\left(\frac{1}{1-\omega}\right) + o(\tau) \\ &= 2\omega\log_2(\omega) + 2(1-\omega)\log_2(1-\omega) + o(\tau) \\ &= -2H(\omega) + o(\tau) \end{aligned}$$

So by plugging this expression with (24) in (23) we have the result.

□

The sublinear case is also relevant to cryptography since several McEliece cryptosystems actually operate at this regime, this is true for the original McEliece system with fixed rate binary Goppa codes [McE78] or with the MDPC-McEliece cryptosystem [MTSB13]. In this regime, [CTS16] showed that all ISD algorithms have the same asymptotic complexity when the number t of errors to correct is equal to $o(n)$ and this is given by:

$$2^{-t\log_2(1-R)(1+o(1))}$$

Let us compare the exponents of statistical decoding and the ISD algorithms when we want to correct a sub-linear error weight. When $t = o(n)$ the complexity we are after is subexponential in the length. The only algorithm finding moderate weight parity-check equations in subexponential time we found is Algorithm 2. It produces parity-check equations of weight $Rn/2$ in amortized time $\tilde{O}(1)$. So with this algorithm, the exponent of statistical decoding is given by $-2\tau\log_2(1-R)$ which is twice the exponent of all the ISDs. We did not conclude for a relative weight $< R/2$ as in any case, all the algorithms we found needed exponential time to output enough equations to perform statistical decoding. So unless one comes up with an algorithm that is able to produce parity-check equations of relative weight $< R/2$ in subexponential time, statistical decoding is not better that any ISDs when we have to correct $t = o(n)$ errors.

5 Studying the single weight case is sufficient

The previous section showed that if it is much more favorable when it comes to perform statistical decoding to produce parity-check equations following the binomial model of weight w rather than parity-checks of constant weight w. The problem is that as far as we know, there is no efficient way of producing moderate weight parity-check equations (let us say that we call moderate any weight $\leq 1 + Rn/2$) which would follow such a model. Even the "easy case", where $w = 1 + Rn/2$ and where it is trivial to produce such equations by simply putting the parity-check matrix in systematic form and taking rows in this matrix [2], does not follow the binomial model : the standard deviation of the parity-check equation weight is easily seen to be different between what is actually produced by the algorithm and the binomial model of weight $1 + Rn/2$. Of course, this does not mean that we should rule out the possibility that there might exist such efficient algorithms. We will however prove that under very mild conditions, that even such an algorithm were to exist then anyway it would produce by nature

[2] For more details see Section 6

parity-checks of different weights and that we would have a statistical decoding algorithm of the same exponential complexity which would keep only *one very specific weight*. In other words, it is sufficient to care about the single weight case as we do here when we study just the exponential complexity of statistical decoding.

To verify this, we fix an arbitrary position we want to decode and assume that some algorithm has produced in time T, $m = \sum_{j=1}^{n} m_j$ parity check equations involving this position where m_j denotes the number of parity-check equations of weight j. The equations of weight j are denoted by $h_1^j, \ldots, h_{m_j}^j$. Statistical decoding is based on simple statistics involving the values $\langle y, h_s^j \rangle$. To simplify a little bit the expressions we are going to manipulate, let us introduce

$$X_s^j \stackrel{\triangle}{=} \langle y, h_s^j \rangle$$

Similarly to Assumptions 1 and 2, we assume that the distribution of $\langle y, h_s^j \rangle$ is approximated by the distribution of $\langle y, h_s^j \rangle$ when h_s^j is drawn uniformly at random among the words of weight j and the $\langle y, h_s^j \rangle$'s are independent. So we have $X_s^j \sim \mathscr{B}(1/2 + \varepsilon_l(j))$ under the hypothesis $\mathscr{H}_l$ and $\varepsilon_l(j)$ is the bias defined in Subsection 3.1 for a weight j. Our aim now is to find a test distinguishing both hypotheses $\mathscr{H}_0$ and $\mathscr{H}_1$. As in Subsection 3.1 it will be the Neymann-Pearson test. We define the following quantity where $\mathbb{P}_{\mathscr{H}_l}$ denotes the probability under the hypothesis $\mathscr{H}_l$:

$$q \stackrel{\triangle}{=} \ln \left(\frac{\mathbb{P}_{\mathscr{H}_0} \left(X_1^1 = x_1^1, \cdots, X_{m_1}^1 = x_{m_1}^1, \cdots, X_1^n = x_1^n, \cdots, X_{m_n}^n = x_{m_n}^n \right)}{\mathbb{P}_{\mathscr{H}_1} \left(X_1^1 = x_1^1, \cdots, X_{m_1}^1 = x_{m_1}^1, \cdots, X_1^n = x_1^n, \cdots, X_{m_n}^n = x_{m_n}^n \right)} \right)$$

The lemma of Neymann-Pearson tells to us to proceed as follows: if $q > \Theta$, where Θ is some threshold, choose $\mathscr{H}_0$ and $\mathscr{H}_1$ otherwise. In this case, no other statistic test will lead to lower false detection probabilities at the same time. In our case, it is enough to set the threshold Θ to 0 since it can be easily verified that no other choices will not change the exponent of the number of samples we need for having vanishing false detection probabilities. We set $p_l(j) \stackrel{\triangle}{=} 1/2 + \varepsilon_l(j)$, $I_0(j) = \#\{0 \in \{x_1^j, \cdots, x_{m_j}^j\}\}$ and $I_1(j) = \#\{1 \in \{x_1^j, \cdots, x_{m_j}^j\}\}$, we have:

$$\frac{\mathbb{P}_{\mathscr{H}_0} \left(X_1^1 = x_1^1, \cdots, X_{m_1}^1 = x_{m_1}^1, \cdots, X_1^n = x_1^n, \cdots, X_{m_n}^n = x_{m_n}^n \right)}{\mathbb{P}_{\mathscr{H}_1} \left(X_1^1 = x_1^1, \cdots, X_{m_1}^1 = x_{m_1}^1, \cdots, X_1^n = x_1^n, \cdots, X_{m_n}^n = x_{m_n}^n \right)} = \prod_{j=1}^{n} \frac{p_0(j)^{I_1(j)} \cdot (1 - p_0(j))^{I_0(j)}}{p_1(j)^{I_1(j)} \cdot (1 - p_1(j))^{I_0(j)}}$$

Therefore by taking the natural logarithm of this expression and $\sum_{k=1}^{m_j} X_k^j = I_1(j)$ and $I_1(j) + I_0(j) = m_j$, we have:

$$\begin{aligned}
q &= \sum_{j=1}^{n} I_0(j) \left[\ln(1 - p_0(j)) - \ln(1 - p_1(j)) \right] + I_1(j) \left[\ln(p_0(j)) - \ln(p_1(j)) \right] \\
&= \sum_{j=1}^{n} (m_j - I_1(j)) \left[\ln(1 - p_0(j)) - \ln(1 - p_1(j)) \right] + \sum_{s=1}^{m_j} X_s^j \left[\ln(p_0(j)) - \ln(p_1(j)) \right] \\
&= \sum_{j=1}^{n} \sum_{s=1}^{m_j} X_s \left[\ln(p_0(j)) - \ln(1 - p_0(j)) + \ln(1 - p_1(j)) - \ln(p_1(j)) \right] \\
&\qquad + m_j \ln \frac{1 - p_0(j)}{1 - p_1(j)}
\end{aligned}$$

We now use the Taylor series expansion around 0 : $\ln(1/2+x) = -\ln(2) + 2x - \frac{4x^2}{2} + \frac{8x^3}{3} + o(x^3)$ and we deduce for i in $\{0,1\}$:

$$\begin{aligned}\ln(p_i(j)) &= \ln(1/2+\varepsilon_i(j)) \\ &= -\ln(2) + 2\varepsilon_i(j) - 2\varepsilon_i(j)^2 + (8/3)\varepsilon_i(j)^3 + o(\varepsilon_i(j)^3)\end{aligned}$$

$$\begin{aligned}\ln(1-p_i(j)) &= \ln(1/2-\varepsilon_i(j)) \\ &= -\ln(2) - 2\varepsilon_i(j) - 2\varepsilon_i(j)^2 - (8/3)\varepsilon_i(j)^3 + o(\varepsilon_i(j)^3)\end{aligned}$$

We have,

$$\begin{aligned}q &= \sum_{j=1}^{n}\sum_{s=1}^{m_j} X_s \cdot \big((4\varepsilon_0(j) + (16/3)\varepsilon_0(j)^3 + o(\varepsilon_0(j)^3) - 4\varepsilon_1(j) - (16/3)\varepsilon_1(j)^3 + o(\varepsilon_1(j)^3)\big) \\ &\qquad - 2m_j \cdot (\varepsilon_0(j) - \varepsilon_1(j) + o(\varepsilon_0(j)) + o(\varepsilon_1(j))) \\ &= 4\sum_{j=1}^{n}\sum_{s=1}^{m_j} X_s^j \left((\varepsilon_0(j) - \varepsilon_1(j) + O(\varepsilon_0(j)^3) + O(\varepsilon_1(j)^3)\right) \\ &\qquad + m_j \ln\frac{1-p_0(j)}{1-p_1(j)} \\ &\approx 4\sum_{j=1}^{n}\sum_{s=1}^{m_j} Y_s^j + C\end{aligned}$$

where

$$Y_s^j \stackrel{\triangle}{=} (\varepsilon_0(j) - \varepsilon_1(j))X_s^j$$

and C is the constant defined by:

$$C \stackrel{\triangle}{=} +m_j \ln\frac{1-p_0(j)}{1-p_1(j)}$$

This computation suggests to use the random variables Y_s^j to build our distinguisher with the Neyman-Pearson likelihood test. By the assumptions on the X_s^j's, the Y_s^j's are independent and we have under $\mathscr{H}_l$:

$$\mathbb{P}\left(Y_s^j = 0\right) = \frac{1}{2} - \varepsilon_l(j) \quad ; \quad \mathbb{P}\left(Y_s^j = (\varepsilon_0(j) - \varepsilon_1(j))\right) = \frac{1}{2} + \varepsilon_l(j)$$

The expectation of Y_s^j under $\mathscr{H}_l$ is given by:

$$\mathbb{E}\left(Y_s^j\right) = (\varepsilon_0(j) - \varepsilon_1(j)) \cdot \left(\frac{1}{2} + \varepsilon_l(j)\right)$$

As for our previous distinguisher we define the random variable V_m for $m = \sum_{j=1}^{n} m_j$ uniform and independent draws of vectors h_s^j in $\mathcal{H}_{w_j,i}$:

$$V_m \stackrel{\triangle}{=} \sum_{j=1}^{n}\sum_{s=1}^{m_j} Y_s^j$$

The expectation of V_m depends on which hypothesis $\mathscr{H}_l$ holds. When hypothesis $\mathscr{H}_l$ holds, we denote the expecation of V_n by E_l. The difference $E_0 - E_1$ is given by:

$$\begin{aligned} E_0 - E_1 &= \sum_{j=1}^{n}\sum_{k=1}^{m_j}(\varepsilon_0(j) - \varepsilon_1(j))\left(\frac{1}{2} + \varepsilon_0(j)\right) - (\varepsilon_0(j) - \varepsilon_1(j))\left(\frac{1}{2} + \varepsilon_1(j)\right) \\ &= \sum_{j=1}^{n}\sum_{k=1}^{m_j}(\varepsilon_0(j) - \varepsilon_1(j))^2 \\ &= \sum_{j=1}^{n} m_j(\varepsilon_0(j) - \varepsilon_1(j))^2 \end{aligned}$$

The deviations of V_m around its expectation will be quantified through Hoeffding's bound which gives in this case up to constant factors in the exponent the right behavior of the probability that V_m deviates from its expectation

Proposition 8 (Hoeffding's Bound)**.** *Let $Y_1, \cdots, Y_m$ independent random variables, $a_1, \cdots, a_m$ and $b_1, \cdots, b_m$ with $a_s < b_s$ such that:*

$$\forall s,\ \mathbb{P}\left(a_s \leq Y_s \leq b_s\right) = 1$$

We set $Z_m = \sum_{s=1}^{m} Y_s$, then:

$$\mathbb{P}\left(|Z_m - \mathbb{E}(Z_m)| \geq t\right) \leq 2\exp\left(-\frac{2t^2}{\sum_{s=1}^{m}(b_s - a_s)^2}\right)$$

In order to distinguish both hypotheses, we set $t = \frac{E_0 - E_1}{2}$. So under $\mathscr{H}_l$, we have

$$\begin{aligned} \mathbb{P}\left(|V_m - E_l| \geq \frac{E_0 - E_1}{2}\right) &= \mathbb{P}\left(|V_m - E_l| \geq \sum_{j=1}^{n}\frac{m_j}{2}(\varepsilon_0(j) - \varepsilon_1(j))^2\right) \\ &\leq 2\exp\left(-\frac{2/4\left(\sum_{j=1}^{n} m_j(\varepsilon_0(j) - \varepsilon_1(j))^2\right)^2}{\sum_{j=1}^{n} m_j(\varepsilon_0(j) - \varepsilon_1(j))^2}\right) \\ &= 2\exp\left(-\frac{1}{2}\left(\sum_{j=1}^{n} m_j(\varepsilon_0(j) - \varepsilon_1(j))^2\right)\right) \end{aligned}$$

We decide that hypothesis $\mathscr{H}_1$ holds if $V_m < \frac{E_0 + E_1}{2}$ and that $\mathscr{H}_0$ holds otherwise. It is clear that the probability P_e to make a wrong decision with this distinguisher is smaller than $2e^{-\frac{1}{2}\left(\sum_{j=1}^{n} m_j(\varepsilon_0(j) - \varepsilon_1(j))^2\right)}$. If we want $P_e \leq 2e^{-\eta}$ for any fixed η, $m_1, \cdots, m_n$ have to be such that:

$$\frac{1}{2}\sum_{j=1}^{n} m_j(\varepsilon_0(j) - \varepsilon_1(j))^2 \geq \eta \Rightarrow \sum_{j=1}^{n} m_j(\varepsilon_0(j) - \varepsilon_1(j))^2 \geq 2\eta \tag{25}$$

Note that this is really the right order (up to some contant factor) for the amount of equations which is needed (the Hoeffding bound captures well up to constant factors the probability of the error of the distinguisher in this case) and using optimal Bayesian decision does not allow to change up to multiplicative factors the number of equations that are needed for a fixed relative error weight. Now assume that

Assumption 3. If we can compute m parity-check equations of weight w in time T, we are able to compute $n \cdot m$ parity-check equations of this weight in time $O(nT)$.

This assumption holds for all "reasonable" randomized algorithms producing random parity-checks with uniform/quasi uniform probability as long as $n \cdot m$ is at most some constant fraction (with a constant < 1) of the total number of parity-check equations. Now we set j_0 such that:

$$m_{j_0}(\varepsilon_0(j_0) - \varepsilon_1(j_0))^2 = \max_{1 \leq j \leq n} \{m_j(\varepsilon_0(j) - \varepsilon_1(j))^2\} \tag{26}$$

Clearly if we take now instead of the original m parity-check equations just the $n \cdot m_{j_0}$ parity check equations of weight j_0 the probability does of error does not get smaller than the bound $2e^{-\eta}$ that we had before since

$$n \cdot m_{j_0}(\varepsilon_0(j_0) - \varepsilon_1(j_0))^2 \geq \sum_{j=1}^{n} m_j(\varepsilon_0(j) - \varepsilon_1(j))^2 \Rightarrow 2e^{-\frac{1}{2}n \cdot m_{j_0}(\varepsilon_0(j) - \varepsilon_1(j))^2} \leq 2e^{-\eta}$$

So, under Assumption 3 if our distinguisher with several weights has enough parity-check equations available, we are able in polynomial time to compute $n \cdot m_{j_0}$ parity-check equations of weight w_{j_0} where j_0 is chosen such that (26) holds and with these parity-check equations the distinguisher of Subsection 3.1 can work too. The complexity of statistical decoding without the phase of computation of the parity-check equations is the number of parity-check equations that it is needed. So, under Assumption 3, its complexity with our first distinguisher will be for each codelength n the same up to a polynomial mutiplicative factor as the complexity with the second distinguisher. Moreover, under Assumption 3 the complexity of the computation of the parity-check equations that is needed for both distinguishers is the same up to a polynomial factor. As the $\varepsilon_1(j) - \varepsilon_0(j)$ are exponentially small in n, in order to have a probability of success which tends to 1, the m_j's of both distinguisher have to be of order $\tilde{O}\left(\frac{1}{(\varepsilon_0(j) - \varepsilon_1(j))^2}\right)$. It leads to the conclusion that the asymptotic exponent of the statistical decoding is the same with considering some well chosen weight or several weights. We stress that this conclusion is about an asymptotic study of the complexity of statistical decoding. Indeed, in practice Algorithms 2 and 3 can output many parity-check equations of weight "close" to $Rn/2$ and $r + (Rn - l)/2$. It will be counter-productive not to keep them and use them with the distinguisher we just described.

6 A simple way of obtaining moderate weight parity-check equations

As we are now able to give a formula for $\pi(\omega, \tau)$ we come back to the algorithm $\texttt{ParityCheckComputation}_w$ in order to estimate $\pi^{complete}(\omega, \tau)$. There is an easy way of producing parity-check equations of moderate weight by Gaussian elimination. This is given in Algorithm 2 that provides a method for finding parity-check equations of weight $w = \frac{Rn}{2}$ of an $[n, Rn]$ random code. Gaussian elimination ($\texttt{GElim}$) of an $Rn \times n$ matrix G_0 consists in finding U ($Rn \times Rn$ and non-singular) such that:

$$UG_0 = [I_{Rn}|G']$$

$L_j(G)$ denotes the $j-$th row of G in Algorithm 2.

Algorithm 2 $\texttt{ParityCheckComputation}_{Rn/2}$

1: Input : $G \in \mathbb{F}_2^{Rn \times n}, i \in \mathbb{N}$
2: Output : $\mathcal{S}_i$ /*$P_{Rn/2}$ *parity-check equations**/
3: $\mathcal{S}_i \leftarrow [\,]$
4: **while** $|\mathcal{S}_i| < P_{Rn/2}$ **do**
5: $P \leftarrow$ random $n \times n$ permutation matrix
6: $[G'|I_{Rn}] \leftarrow \texttt{GElim}(GP)$ and if it fails return to line 5
7: $H \leftarrow [I_{n(1-R)}|{G'}^T]$ /*Parity matrix of the code*/
8: **for** $j = 1$ to $n(1-R)$ **do**
9: **if** $L_j(H)_i = 1$ and $w_H(L_j(H)) = Rn/2$ **then**
10: $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{L_j(H)P^T\}$
11: **end if**
12: **end for**
13: **end while**
14: **return** $\mathcal{S}$

Algorithm 2 is a randomized algorithm. Randomness comes from the choice of the permutation P. It is straightforward to check that this algorithm returns $P_{Rn/2}$ parity-check equations of weight $Rn/2$ in time $\tilde{O}\left(P_{Rn/2}\right)$.

Now we set $\tau = H^{-1}(1-R)$. This relative weight, which corresponds to the Gilbert-Varshamov bound, is usually used to measure the efficiency of decoding algorithms. Indeed it corresponds to the critical error weight below which we still have with probability $1 - o(1)$ a unique solution to the decoding problem. It can be viewed as the weight for which the decoding problem is the hardest, since the larger the weight the more difficult the decoding problem seems to be (this holds at least for all known decoding algorithms of generic linear codes). As a consequence of Propositions 2 and 4, we have the following theorem:

Theorem 3. *[Naive Statistical Decoding's asymptotic complexity]*
With the computation of parity-check equations of weight $Rn/2$ thanks to $\texttt{ParityCheckComputation}_{Rn/2}$*, we have:*

$$\pi(R/2, \tau) = \pi^{complete}(R/2, \tau)$$

where $\pi(R/2, \tau)$ is given by Theorem 2.

Exponents (as a function of R) of Prange's ISD and statistical decoding are given in Figure 5. As we see the difference is huge. This version of statistical decoding can not be considered as an improvement over ISDs. However, as $\omega \mapsto \pi(\omega, \tau)$ for τ fixed is an increasing function in ω, we have to study the case $\omega < R/2$. It is the subject of the next section. We will give there an algorithm computing efficiently parity-check equations of smaller weight than $Rn/2$. However we also prove there that no matter how efficiently we perform the pre-computation step, any version of statistical decoding is worse than Prange's ISD.

7 Improvements and limitations of statistical decoding

7.1 Framework

Before giving an improvement and giving lower bounds on the complexity of statistical decoding, we would like to come back to the computation problem of the $\mathcal{S}_i$'s in the complexity of

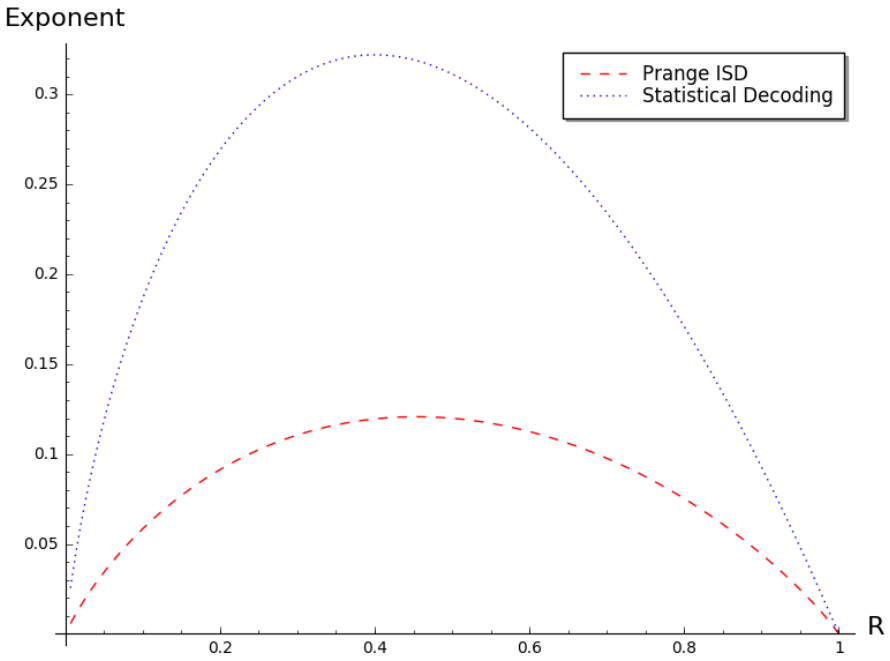

Figure 5: Asymptotic Exponents of Prange ISD and Statistical Decoding for $\tau = H^{-1}(1-R)$ et $\omega = R/2$

statistical decoding. Our aim is to clarify the picture a little bit. We stress that statistical decoding complexity is, if the $\mathcal{S}_i$'s are already computed and stored, (up to a polynomial factor) the number of equations we use to take our decision. We denote by $\mathscr{D}_w$ the part of statistical decoding which uses these parity-check equations to perform the decoding and by $\mathscr{A}_w$ the randomized algorithm used for outputting a certain number of random parity-check equations of weight w. $\texttt{ParityCheckComputation}_w$ is assumed to make a certain number of calls to $\mathscr{A}_w$. It is assumed that $\mathscr{A}_w$ outputs N_w parity-check equations of weight w in time T_w each time we run it. We assume that statistical decoding needs $\tilde{O}(P_w)$ equations. If we consider the computations of parity-check equations as part of statistical decoding, its complexity is given by:

$$\tilde{O}\left(P_w + T_w \cdot \max(1, \frac{P_w}{N_w})\right)$$

When $\frac{T_w}{N_w} = \tilde{O}(1)$, we say $\mathscr{A}_w$ gives equations in amortized time $\tilde{O}(1)$. With this condition if we assume $P_w \geq N_w$, the complexity is the number of equations needed.

In any case, complexity of statistical decoding is lower-bounded by $\tilde{O}(P_w)$ and the lower the equation weight w, the lower the number of equations P_w we need for performing statistical decoding. The goal of this section is to show how to find many parity-check equations of weight $< Rn/2$ in an efficient way and to give a minimal weight for which it makes sense to make this operation.

7.2 A lower bound on the complexity of statistical decoding

As we just pointed out, statistical decoding needs $\tilde{O}(P_w)$ parity-check equations of weight w to work. Its complexity is therefore always greater than $\tilde{O}(P_w)$. We assume again the code

we want to decode to be a random code. This assumption is standard in the cryptographic context. The expected number of parity-check equations of weight w in an $[n, Rn]$ random binary linear code is $\frac{\binom{n}{w}}{2^{Rn}}$. Obviously if w is too small there are not enough equations for statistical decoding to work, we namely need that

$$P_w \leq \frac{\binom{n}{w}}{2^{Rn}}.$$

The minimum $\omega_0(R, \tau)$ such that this holds is clearly given by the minimal ω such that the following expression holds

$$\pi(\omega, \tau) = H(\omega) - R$$

So $\omega_0(R, \tau)$ gives the minimal relative weight such that asymptotically the number of parity-check equations needed for decoding is exactly the number of parity-check equations of weight $w_0(R, \tau)$ in the code, where $w_0(R, \tau) \stackrel{\triangle}{=} \omega_0(R, \tau)n$. Below this weight, statistical decoding can not work (at least not for random linear codes). In other words the asymptotic exponent of statistical decoding is always lower-bounded by $\pi(w_0(R, \tau), \tau)$.

In the case of a relative error weight given by the Gilbert-Varshamov bound $\tau_{\mathrm{DGV}} = H^{-1}(1 - R)$, Theorem 3 leads to the conclusion that

$$\omega_0(R, \tau_{\mathrm{DGV}}) = \frac{1}{2} - \sqrt{\tau_{\mathrm{DGV}} - \tau_{\mathrm{DGV}}^2}$$

Moreover for all relative weights greater than $\omega_0(R, \tau_{\mathrm{DGV}})$ the number of parity-check equations that are needed is exactly the number of parity-check equations of this weight that exist in a random code. This result is rather intriguing and does not seem to have a simple interpretation. The relative minimal weight $w_0(R, \tau_{\mathrm{DGV}})$ is in relationship with the first linear programming bound of McEliece-Rodemich-Rumsey-Welch and can be interpreted through its relationship with the zeros of Krawtchouk polynomials. This bound arises from the fact that from Theorem 3, we know that $\omega_0(R, \tau_{\mathrm{DGV}})$ corresponds to the relative weight where we switch from the complex case to the real case, and this happens precisely when we leave the region of zeros of the Krawtchouk polynomials.

Thanks to Figure 6 which compares Prange's ISD, statistical decoding with parity-check equations of relative weight $R/2$ and $\omega_0(R, \tau)$ with $\tau = H^{-1}(1 - R)$, we clearly see on the one hand that there is some room of improving upon naive statistical decoding based on parity-check equations of weight $Rn/2$, but on the other hand that even with the best improvement upon statistical decoding we might hope for, we will still be above the most naive information set decoding algorithm, namely Prange's algorithm.

7.3 An improvement close to the lower bound

The goal of this subsection is to present an improvement to the computation of parity-check equations and to give its asymptotic complexity. R. Overbeck in [Ove06, Sec. 4] showed how to compute parity-check equations thanks to Stern's algorithm. We are going to use this algorithm too. However, whereas Overbeck used many iterations of this algorithm to produce a few parity-check equations of small weight, we observe that this algorithm produces in a natural way during its execution a large number of parity-check equations of relative weight smaller than $R/2$. We will analyze this process here and show that it yields an algorithm $\mathscr{A}_w$ that gives equations in amortized time $\tilde{O}(1)$.

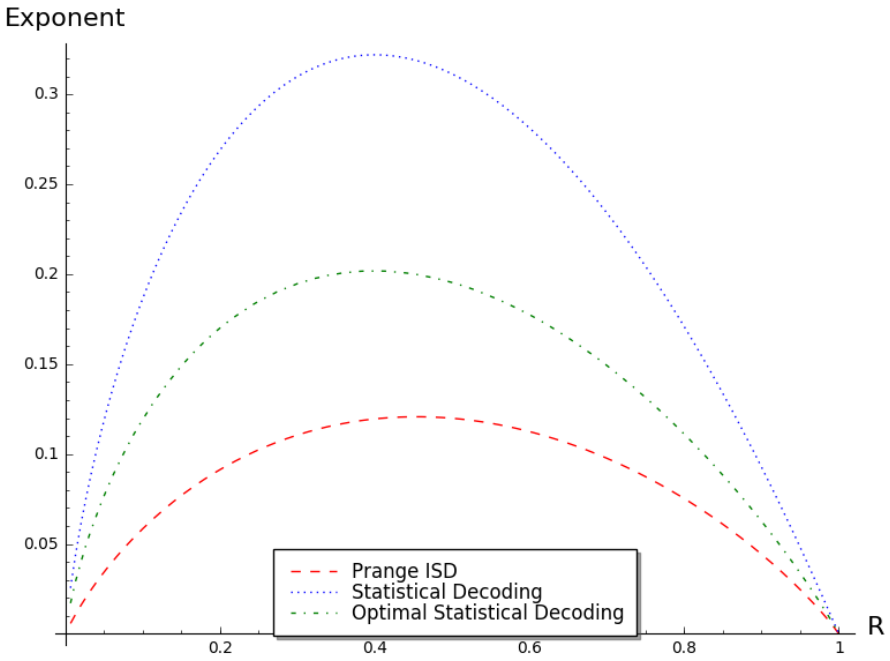

Figure 6: Asymptotic exponents of Prange ISD, naive statistical decoding and optimal/optimistic statistical decoding for $\tau = H^{-1}(1-R)$

To find parity-check equations, we described an algorithm which just performs Gaussian elimination and selection of sufficiently sparse rows. In fact, it is the main idea of Prange's algorithm. As we stressed in introduction, this algorithm has been improved rather significantly over the years (ISD family). Our idea to improve the search for parity-check equations is to use precisely these improvements. The first significant improvement is due to Stern and Dumer [Ste88, Dum91]. The main idea is to solve a sub-problem with the birthday paradox. We are going to describe this process and show how it allows to improve upon naive statistical decoding.

We begin by choosing a random permutation matrix $P \in \mathbb{F}_2^{n\times n}$ and putting the matrix GP into the systematic form:

$$\begin{bmatrix} I_{Rn-l} & G_1 \\ 0 & G_2 \end{bmatrix} \text{ where } G_1 \in \mathbb{F}_2^{(Rn-l)\times(n(1-R)+l)} \text{ and } G_2 \in \mathbb{F}_2^{l\times(n(1-R)+l)}$$

1. We solve $\mathrm{CSD}(G_2, r, 0_{[l]})$.
2. For each solution e, we output $e_s = (eG_1^T, e)P^T$.

Remark 3. We recall that solving $\mathrm{CSD}(G_2, r, 0_{[l]})$ means to find r columns of G_2 which yield 0.

· **Soundness:** We have

$$Ge_s^T = GP\begin{bmatrix} G_1e^T \\ e^T \end{bmatrix} = \begin{bmatrix} I_{Rn-l} & G_1 \\ 0 & G_2 \end{bmatrix}\begin{bmatrix} G_1e^T \\ e^T \end{bmatrix} = \begin{bmatrix} G_1e^T + G_1e^T \\ G_2e^T \end{bmatrix} = 0$$

and therefore e is a parity-check equation of $\mathscr{C}$.

· **Number of solutions:** The number of solutions is given by the number of solutions of 1. Furthermore, the complexity of this algorithm is up to a polynomial factor given by the complexity of 1.

Remark 4. This algorithm may not provide in one step enough solutions. In this case, we have to put G in another systematic form (*i.e.* choose another permutation). The randomness of our algorithm will come from this choice of permutation matrix.

· **Solutions' weight:** In our model G is supposed to be random. So we can assume the same hypothesis for G_2. As the length of its rows is $Rn-l$, we get asymptotically parity-check equations of weight:

$$r + \frac{Rn-l}{2}(1+o(1))$$

The first part of this algorithm can be viewed as the first part of ISD algorithms. There is a general presentation of these algorithms in [FS09] in Section 3. All the efforts that have been spent to improve Prange's ISD can be applied to solve the first point of our algorithm. To solve this point, Dumer suggested to put G_2 in the following form:

$$G_2 = [G_2^{(1)}|G_2^{(2)}] \text{ where } G_2^{(i)} \in \mathbb{F}_2^{l\times\frac{n(1-R)+l}{2}}$$

and to build the lists:

$$\mathscr{L}_1 = \left\{ \left(e_1, G_2^{(1)}e_1^T\right) \mid e_1 \in \mathbb{F}_2^{\frac{n(1-R)+l}{2}} \text{ and } w_H(e_1) = \frac{r}{2} \right\}$$

$$\mathscr{L}_2 = \left\{ \left(e_2, G_2^{(2)}e_2^T\right) \mid e_2 \in \mathbb{F}_2^{\frac{n(1-R)+l}{2}} \text{ and } w_H(e_2) = \frac{r}{2} \right\}$$

Then we intersect these two lists with respect to the second coordinate and we keep the associated first coordinate. In other words, we get:

$$\{(e_1,e_2) \mid w_H(e_i) = \frac{r}{2} \text{ and } G_2^{(1)}e_1^T = G_2^{(2)}e_2^T\}$$

Remark 5. This process is called a fusion.

Algorithm 3 summarizes this formally.

Algorithm 3 `DumerFusion`

Input : $G \in \mathbb{F}_2^{Rn \times n}, l, r$.
Output : $\mathcal{S}$ /*subset of $\mathcal{H}_w$*/
$\mathcal{S} \leftarrow [\,]$ /*Empty list*/
$\mathcal{T} \leftarrow [\,]$ /* Hash table*/
$P \leftarrow$ random $n \times n$ permutation matrix
We find $U \in \mathbb{F}_2^{Rn \times Rn}$ non-singular such that $UGP = \begin{bmatrix} I_{Rn-l} & G_1 \\ 0 & G_2 \end{bmatrix}$
We partition G_2 as $[G_2^{(1)}|G_2^{(2)}]$ where $G_2^{(i)} \in \mathbb{F}_2^{l \times \binom{n(1-R)+l}{2}}$
for all $e_1 \in \mathbb{F}_2^{(n(1-R)+l)/2}$ of weight $r/2$ **do**
    $x \leftarrow G_2^{(1)} e_1^T$
    $\mathcal{T}[x] \leftarrow \mathcal{T}[x] \cup \{e_1\}$
end for
for all $e_2 \in \mathbb{F}_2^{(n(1-R)+l)/2}$ of weight $r/2$ **do**
    $x \leftarrow G_2^{(2)} e_2^T$
    for all $e_1 \in \mathcal{T}[x]$ **do**
        $e \leftarrow (e_1, e_2)$
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{(eG_1^T, e)P^T\}$
    end for
end for

As we neglect polynomial factors, the complexity of Algorithm 3. is given by:

$$\tilde{O}\left(\binom{(n(1-R)+l)/2}{r/2} + \#\mathcal{S}\right)$$

Indeed, we only have to enumerate the hash table construction (first factor) and the construction of $\mathcal{S}$. In order to estimate $\#\mathcal{S}$ we use the following classical proposition:

Proposition 9. *Let $L_1, L_2 \subseteq \{0,1\}^l$ be two lists where inputs are supposed to be random and distributed uniformly. Then, the expectation of the cardinality of their intersection is given by:*

$$\frac{\#L_1 \cdot \#L_2}{2^l}$$

As we supposed G_2 random, we can apply this proposition to `DumerFusion`. Therefore,

Proposition 10 (`DumerFusion`'s complexity)**.**
`DumerFusion`*'s complexity is given by:*

$$\tilde{O}\left(\binom{(n(1-R)+l)/2}{r/2} + \frac{\binom{(n(1-R)+l)/2}{r/2}^2}{2^l}\right)$$

and it provides on average

$$\frac{\binom{(n(1-R)+l)/2}{r/2}^2}{2^l}$$

solutions

In order to study this algorithm asymptotically, we introduce the following notations and relative parameters:

Notation 6.
- $N_{r,l} \triangleq \frac{\binom{(n(1-R)+l)/2}{r/2}^2}{2^l}$;
- $T_{r,l} \triangleq \binom{(n(1-R)+l)/2}{r/2} + \frac{\binom{(n(1-R)+l)/2}{r/2}^2}{2^l}$;
- $\rho = \frac{r}{n}$;
- $\lambda = \frac{l}{n}$.

We may observe that $N_{r,l}$ gives the number of parity-check equations that `DumerFusion` outputs in one iteration and $T_{r,l}$ is the running time of one iteration. There are many ways of choosing r and l. However in any case (see Subsection 7.2), as the weight of parity-check equations we get with `DumerFusion` is $(r + \frac{R-l}{2})(1 + o(1))$ we have to choose r and l such that

$$w_0(R, t) \leq r + (R - l)/2$$

which is equivalent to

$$\omega_0(R, \tau) \leq \rho + \frac{R - \lambda}{2} \tag{27}$$

The following lemma gives an asymptotic choice of ρ and λ that allows to get parity-check equations in amortized time $\tilde{O}(1)$:

Lemma 11. *If*

$$\rho = (1 - R + \lambda) \cdot H^{-1}\left(\frac{2\lambda}{1 - R + \lambda}\right) \tag{28}$$

`DumerFusion` *provides parity-check equations of relative weight* $\rho + \frac{R-\lambda}{2}$ *in amortized time* $\tilde{O}(1)$. *Moreover, with this constraint we have asymptotically :*

$$N_{r,l} = \tilde{O}\left(2^{\lambda \cdot n}\right)$$

Proof. We remark that $T_{r,l} = N_{r,l} + \binom{(n(1-R)+l)/2}{r/2}$. Our goal is to find ρ, λ such that asymptotically $\frac{T_{r,l}}{N_{r,l}} = \tilde{O}(1)$. The constraint (28) follows from $\binom{u}{v} = \tilde{O}\left(2^{u \cdot H(u/v)}\right)$.

□

We are now able to give the asymptotic complexity of statistical decoding with the use of `DumerFusion` strategy.

Theorem 4. *With the constraints (27), (28) and*

$$\lambda \leq \pi(\rho + \frac{R - \lambda}{2}, \tau) \tag{29}$$

for (ρ, λ) *we have:*

$$\pi^{complete}(\rho + (R - \lambda)/2, \tau) = \pi(\rho + (R - \lambda)/2, \tau)$$

Proof. Thanks to (28) and (29) we use Subsection 7.1 and we conclude that under theses constraints we have $\pi(\rho + (r - \lambda)/2, \tau) = \pi^{complete}(\rho + (r - \lambda)/2, \tau)$. □

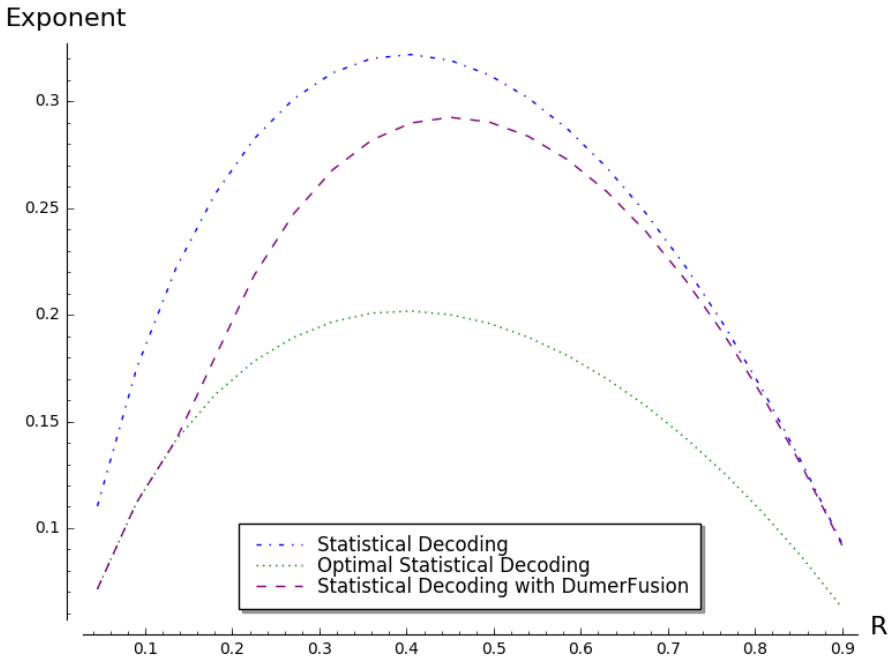

Figure 7: Asymptotic exponents of naive statistical decoding and with the use of optimal `DumerFusion` and optimal/optimistic statistical decoding for $\tau = H^{-1}(1-R)$

Remark 6. We summarize the meaning of the constraints as:

- With (27) we are sure there exists enough parity-check equations for statistical decoding to work;
- With (28) `DumerFusion` gives parity-check equations in amortized time $\tilde{O}(1)$;
- With (29) `DumerFusion` provides always no more equations in one iteration than we need.

In order to get the optimal statistical decoding complexity we minimize $\pi(\rho+(R-\lambda)/2,\tau)$ (with $\pi(\rho+(R-\lambda)/2,\tau)$ given by Theorem 2) under constraints (27), (28) and (29). The exponent of statistical decoding with this strategy is given in Figure 7.

As we see, `DumerFusion` with our strategy allows statistical decoding to be optimal for rates close to 0. We can further improve `DumerFusion` with ideas of [MMT11] and [BJMM12], however this comes at the expense of having a much more involved analysis and would not allow to go beyond the barrier of the lower bound on the complexity of statistical decoding given in the previous subsection. Nevertheless with the same strategy, these improvements lead to better rates with an optimal work of statistical decoding.

8 Conclusion

In this article we have revisited statistical decoding with a rigorous study of its asymptotic complexity. We have shown that under Assumption 1 and 2 this algorithm is regardless of any strategy we choose for producing the moderate weight parity-check equations needed

by this algorithm always worse than Prange ISD for the hardest instance of decoding (i.e. for a number of errors equal to Gilbert Varshamov bound). In this case a very intriguing phenomenon happens, we namely need for a large range of parity-check weights all the parity-check available in the code to be be able to decode with this technique. It seems very hard to come up with choices of rate, error weight and length for which statistical decoding might be able to compete with ISD even if this can not be totally ruled out by the study we have made here. However there are clearly more sophisticated techniques which could be used to improve upon statistical decoding. For instance using other strategies by grouping positions together and using all parity-check equations involving bits in this group could be another possible interesting generalization of statistical decoding.

References

[Bar97] Alexander Barg. Complexity issues in coding theory. *Electronic Colloquium on Computational Complexity*, October 1997.

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In *Advances in Cryptology - EUROCRYPT 2012*, Lecture Notes in Comput. Sci. Springer, 2012.

[CTS16] Rodolfo Canto-Torres and Nicolas Sendrier. Analysis of information set decoding for a sub-linear error weight. In *Post-Quantum Cryptography 2016*, Lecture Notes in Comput. Sci., pages 144–161, Fukuoka, Japan, February 2016.

[Dum91] Ilya Dumer. On minimum distance decoding of linear codes. In *Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory*, pages 50–52, Moscow, 1991.

[FKI07] Marc P. C. Fossorier, Kazukuni Kobara, and Hideki Imai. Modeling bit flipping decoding based on nonorthogonal check sums with application to iterative decoding attack of McEliece cryptosystem. *IEEE Trans. Inform. Theory*, 53(1):402–411, 2007.

[FS09] Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In M. Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Comput. Sci.*, pages 88–105. Springer, 2009.

[IS98] Mourad E.H. Ismail and Plamen Simeonov. Strong asymptotics for Krawtchouk polynomials. *Journal of Computational and Applied Mathematics*, pages 121–144, 1998.

[Jab01] Abdulrahman Al Jabri. A statistical decoding algorithm for general linear block codes. In Bahram Honary, editor, *Cryptography and coding. Proceedings of the 8^{th} IMA International Conference*, volume 2260 of *Lecture Notes in Comput. Sci.*, pages 1–8, Cirencester, UK, December 2001. Springer.

[McE78] Robert J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44.

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $O(2^{0.054n})$. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Comput. Sci.*, pages 107–124. Springer, 2011.

[MO15] Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Comput. Sci.*, pages 203–228. Springer, 2015.

[MTSB13] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In *Proc. IEEE Int. Symposium Inf. Theory - ISIT*, pages 2069–2073, 2013.

[Ove06] Raphael Overbeck. Statistical decoding revisited. In Reihaneh Safavi-Naini Lynn Batten, editor, *Information security and privacy : 11th Australasian conference, ACISP 2006*, volume 4058 of *Lecture Notes in Comput. Sci.*, pages 283–294. Springer, 2006.

[Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.

[Ste88] Jacques Stern. A method for finding codewords of small weight. In G. D. Cohen and J. Wolfmann, editors, *Coding Theory and Applications*, volume 388 of *Lecture Notes in Comput. Sci.*, pages 106–113. Springer, 1988.

[SV04] Nandakishore Santhi and Alexander Vardy. On the effect of parity-check weights in iterative decoding. In *Proc. IEEE Int. Symposium Inf. Theory - ISIT*, page 101, Chicago, IL, June 2004.