Algorithmic Coding Theory

Instructor: Amit Kumar Sinhababu

Scribe: Soham Chatterjee

sohamchatterjee999@gmail.com
Website: sohamch08.github.io

2023

Contents

CHAPTER 1

Locally Decodable Codes

1.1 Introduction

References for this topic are [Yek12]

Definition 1.1.1 (Locally Decodable Codes)**.** *A q-ary code $C : \mathbb{F}_q^k \to \mathbb{F}_q^N$ is said to be (r, δ, ϵ)-locally decodable if there exists a randomized decoding algorithm $\mathcal{A}$ such that*

1. *For all $\overline{x} \in \mathbb{F}_q^k$, $i \in [k]$ and all vectors $\overline{y} \in \mathbb{F}_q^N$ such that $\Delta(C(\overline{x}), \overline{y}) \leq \delta$:*

$$Pr[\mathcal{A}^{\overline{y}}(i) = \overline{x}(i)] \geq 1 - \epsilon$$

   where the probability is taken over the random coin tosses of the alforithm $\mathcal{A}$

2. *$\mathcal{A}$ makes at most r queries to $\boldsymbol{y}$*

We would like to have LDCs that for a given message length k and algphabet size q have small values of r, N and ϵ and a large value of δ. The exact value of r is not very important provided that it is much smaller than k. Similarly the exact value of $\epsilon < \frac{1}{2}$ is not the important since one can easily amplify ϵ to be close to 0 by running the decoding procedure few times and taking a majority vote.

A locally decodable code allows to probabilistically decode any coordinate of a message by probing only few coordinates of its corrupted encoding. A stronger property that is desirable in certain application is that of local correctability allowing to efficiently recover not only coordinates of the message but also arbitrary coordinates of the encoding.

Definition 1.1.2 (Locally Correctable Codes)**.** *A q-ary code C in the space $\mathbb{F}_q^N$ is (r, δ, ϵ)-locally decodable if there exists a randomized decoding algorithm $\mathcal{A}$ such that*

1. *For all $\overline{c} \in C$, $i \in [N]$ and all vectors $\overline{y} \in \mathbb{F}_q^N$ such that $\Delta(\overline{c}, \overline{y}) \leq \delta$:*

$$Pr[\mathcal{A}^{\overline{y}}(i) = \overline{c}(i)] \geq 1 - \epsilon$$

   where the probability is taken over the random coin tosses of the algorithm $\mathcal{A}$

2. *$\mathcal{A}$ makes at most r queries to $\boldsymbol{y}$*

Lemma 1.1.1. *Let q be a prime power. Suppose $C \subseteq \mathbb{F}_q^N$ is a (r, δ, ϵ)-locally correctable code that is a linear subspace; then there exists a $q-$ary (r, δ, ϵ)-locally decodable code C' encoding messages of length* $\dim C$ *to codewords of length N*

Proof: Let $I \subseteq [N]$ be a set of $k := \dim C$ coordinates of C whose values uniquely determine an element of C. For $\boldsymbol{c} \in C$ let $\boldsymbol{c}|_I \in \mathbb{F}_q^k$ denote the restriction of $\boldsymbol{c}$ to coordinates of I. Given a message $\boldsymbol{x} \in \mathbb{F}_q^k$ we define $C'(\boldsymbol{x})$ to be the unique element $\boldsymbol{c} \in C$ such that $\boldsymbol{c}|_I = \boldsymbol{x}$. Now C' is a (r, δ, ϵ)-locally decodable code ■

1.2 Reed Muller Locally Decodable Codes

The key idea behind early locally decodable codes is that of polynomial interpolation. Local decodability is achived through reliance on the rich structure of short local dependencies between such evaluations at multiple points. We consider three local corrector for RM codes of increasing level of sophistication.

1.2.1 Basic Decoding on Lines

To recover the value of a degree d polynomial $f \in \mathbb{F}_q[x_1 \ldots, x_n]$ at a point $\boldsymbol{w} \in \mathbb{F}_q^n$ it shoots a random affine line through $\boldsymbol{w}$ and then relies on the local dependency between the calles of f at some $d+1$ points along the line.

1.2.2 Improved Decoding on Lines

1.2.3 Decoding On Curves

We require d is substantially smaller than q. The corrector shoots a random parametric degree 2 curve through $\overline{w}$ and then relies on the high redundancy among the values of F along the curve to recover the value of a degree d polynomial $F \in \mathbb{F}_q[x_1, \ldots, x_n]$.

The advantage upon the decoder of (improved decoding) is that points on a random degree 2 curve constitute a two-d=independent sample from the underlying space.

Theorem 1.2.1. *Let $\sigma < 1$ be a positive real. Let n and d be positive integers. Let q be prime power such that $d \leq \sigma(q-1) - 1$; then there exists a linear code of dimension $k = \binom{n+d}{d}$ in $\mathbb{F}_q^N$, $N = q^n$, that for all positive $\delta < \frac{1}{2} - \sigma$ is $\left(q-1, \delta, O_{\sigma,\delta}\left(\frac{1}{q}\right)\right)$-locally correctable.*

Proof: Given a δ corrupted evaluation of a degree d polynomial F and a point $\overline{w} \in \mathbb{F}_q^n$ the corrector picks vectors $\overline{v}_1, \overline{v}_2 \in \mathbb{F}_q^n$ uniformly at random and considers a degree two curve

$$C = \{\overline{w} + \lambda\overline{v}_1 + \lambda^2\overline{v}_2 \mid \lambda \in \mathbb{F}_q\}$$

through $\overline{w}$. The corrector tries to reconstruct a restriction of F to C which is a polynomial of degree up to $2d$.

The corrector queries coordinates of the evaluation vector corresponding to points $C(\lambda) = \overline{w} + \lambda\overline{v}_1 + \lambda^2\overline{v}_2$, for all $\lambda \in \mathbb{F}_q^*$ to obtain values $\{e_\lambda\}$. It then recovers the unique univariate polynomial $h = F(\overline{w} + \lambda\overline{v}_1 + \lambda^2\overline{v}_2)$, $\deg h \leq 2d$ such that $h(\lambda) = e_\lambda$ for all but at most $\left\lfloor \frac{(1-2\sigma)(q-1)}{2} \right\rfloor$ values of $\lambda \in \mathbb{F}_q^*$ and outputs $h(0)$ by Barlekamp-Welch Algorithm since

$$q - 1 - \frac{(1-2\sigma)(q-1)}{2} = (q-1)\left(1 - \frac{1-2\sigma}{2}\right) = \sigma(q-1) > d$$

Now we will calculate the probability of number of corrupted queries is at most $\left\lfloor \frac{(1-2\sigma)(q-1)}{2} \right\rfloor$. For $\overline{a} \in \mathbb{F}_q^n$ and $\lambda \in \mathbb{F}_q^*$ consider the random variable $x_{\overline{a}}^{\lambda}$ which is the indicator variable of the event $C(\lambda) = \overline{a}$. Let $E \subseteq \mathbb{F}_q^n$

such that $|E| \leq \delta N$ be the set of $\overline{a} \in \mathbb{F}_q^n$ such that the values of F at $\overline{a}$ are corrupted. For every $\lambda \in \mathbb{F}_q^*$ consider a random variable

$$x^\lambda = \sum_{\overline{a} \in E} x_{\overline{a}}^\lambda$$

Note the variables $\{x^\lambda\}$. for all $\lambda \in \mathbb{F}_q^*$ are pairwise independent. For every $\lambda \in \mathbb{F}_q^*$ we have

$$\mathbb{E}\left[x^\lambda\right] \leq \delta \text{ and } \mathrm{Var}\left[x^\lambda\right] = \mathbb{E}\left[\left(x^\lambda\right)^2\right] - \mathbb{E}\left[x^\lambda\right]^2 \leq \delta - \delta^2$$

Now consider the random variable $x = \sum_{\lambda \in \mathbb{F}_q^*} x^\lambda$. Since $\{x^\lambda\}$ are pariwise independent we have

$$\mathrm{Var}[x] = \mathrm{Var}\left[\sum_{\lambda \in \mathbb{F}_q^*} x^\lambda\right] = \sum_{\lambda \in \mathbb{F}_q^*} \mathrm{Var}[x^\lambda]$$

Therefore we have

$$\mathbb{E}[x] = \sum_{\lambda \in \mathbb{F}_q^*} \mathbb{E}\left[x^\lambda\right] \leq (q-1)\delta \text{ and } Var[x] \leq (q-1)(\delta - \delta^2)$$

Therefore

$$\begin{aligned}
Pr\left[x \geq \left\lfloor \frac{(1-2\sigma)(q-1)}{2} \right\rfloor\right] &= Pr\left[x - \mathbb{E}[x] \geq \left\lfloor \frac{(1-2\sigma)(q-1)}{2} \right\rfloor - \mathbb{E}[x]\right] \\
&\leq Pr\left[|x - \mathbb{E}[x]| \geq \left\lfloor \frac{(1-2\sigma)(q-1)}{2} \right\rfloor - \delta(q-1)\right] \\
&\leq Pr\left[|x - \mathbb{E}[x]| \geq \frac{(q-1)(1-2(\sigma+\delta))}{2}\right] \\
&\leq \frac{(q-1)(\delta - \delta^2)}{\left[\frac{(q-1)(1-2(\sigma+\delta))}{2}\right]^2} && \text{[By Theorem 4.1.2]} \\
&= \frac{4(\delta - \delta)}{(q-1)(1-2(\sigma+\delta))^2} = O_{\sigma,\delta}\left(\frac{1}{q}\right)
\end{aligned}$$

Hence we probability $1 - O_{\sigma,\delta}\left(\frac{1}{q}\right)$ we can obtain the correct h and decode the value of F at $\overline{w}$. Therefore it is $\left(q-1, \delta, O_{\sigma,\delta}\left(\frac{1}{q}\right)\right)$-locally correctable. ■

CHAPTER 2

Multiplicity Code

References for this topic are [KSY11], [Kop15]
Notation:

- For a vector $\bar{i} = \langle i_1, i_2, \ldots, i_m \rangle$ of non-negative integers its ***weight*** denoted $wt(\bar{i}) := \sum\limits_{j=1}^{m} i_j$
- $\mathbb{F}[\overline{X}] = \mathbb{F}[X_1, \ldots, X_m]$
- For a vector of non-negative integers $\bar{i}$, $\overline{X}^{\bar{i}} := \prod\limits_{j=1}^{m} X_j^{i_j}$
- $\Delta(x, y) = \underset{i \in [n]}{Pr}[x_i \neq y_i]$

2.1 Hasse Derivative

Definition 2.1.1 ((Hasse) Derivative)**.** *For $P(\overline{X}) \in \mathbb{F}[\overline{X}]$ and non-negative vector $\bar{i}$, the $\bar{i}$th (Hasse) derivative of P denoted $P^{(\bar{i})}(\overline{X})$ is the coefficient of $\overline{Z}^{\bar{i}}$ in the polynomial $\widetilde{P}(\overline{X}, \overline{Z}) \triangleq P(\overline{X} + \overline{Z}) \in \mathbb{F}[\overline{X}, \overline{Z}]$. Thus*

$$P(\overline{X} + \overline{Z}) = \sum_{\bar{i}} P^{(\bar{i})}(\overline{X})\overline{Z}^{\bar{i}}$$

2.1.1 Basic Properties of Hasse Derivatives

Proposition 2.1.1 ([?],[DKSS09])**.** *Let $P(\overline{X}), Q(\overline{X}) \in \mathbb{F}[\overline{X}]$ and let $\bar{i}, \bar{j}$ be vectors of nennegative integers. Then:*

1. $P^{(\bar{i})}(\overline{X}) + Q^{(\bar{i})}(\overline{X}) = (P + Q)^{(\bar{i})}(\overline{X})$
2. $(P \cdot Q)^{(\bar{i})}(\overline{X}) = \sum\limits_{0 \leq \bar{e} \leq \bar{i}} P^{(\bar{e})}(\overline{X}) \cdot Q^{(\bar{i} - \bar{e})}(\overline{X})$
3. $\left(P^{(\bar{i})}\right)^{(\bar{j})}(\overline{X}) = \binom{\bar{i} + \bar{j}}{\bar{i}} P^{(\bar{i} + \bar{j})}(\overline{X})$

Proof:

-
-
- We will expand $P(\overline{X}+\overline{Z}+\overline{W})$ in two ways.

$$P(\overline{X}+(\overline{Z}+\overline{W})) = \sum_{\overline{k}} P^{(\overline{k})}(\overline{X})(\overline{Z}+\overline{W})^{\overline{k}} = \sum_{\overline{k}} P^{(\overline{k})}(\overline{X}) \sum_{\overline{i}+\overline{j}=\overline{k}} \binom{\overline{k}}{\overline{i}} \overline{Z}^{\overline{j}}\overline{W}^{\overline{i}} = \sum_{\overline{i},\overline{j}} P^{(\overline{i}+\overline{j})}(\overline{X}) \binom{\overline{i}+\overline{j}}{\overline{i}} \overline{Z}^{\overline{j}}\overline{W}^{\overline{i}}$$

$$P((\overline{X}+\overline{Z})+\overline{W}) = \sum_{\overline{i}} P^{(\overline{i})}(\overline{X}+\overline{Z})\overline{W}^{\overline{i}} = \sum_{\overline{i}}\sum_{\overline{j}} \left(P^{(\overline{i})}\right)^{(\overline{j})}(\overline{X})\overline{Z}^{\overline{j}}\overline{W}^{\overline{i}}$$

Hence comparing the coefficients of $\overline{Z}^{\overline{j}}\overline{W}^{\overline{i}}$ we obtain $\left(P^{(\overline{i})}\right)^{(\overline{j})}(\overline{X}) = \binom{\overline{i}+\overline{j}}{\overline{i}} P^{(\overline{i}+\overline{j})}(\overline{X})$

■

2.2 Multiplicity

Now we will define the notion of the multiplicity of a polynomial.

Definition 2.2.1 (Multiplicity). *For $P(\overline{X}) \in \mathbb{F}[\overline{X}]$ and $\overline{a} \in \mathbb{F}^m$ the multiplicity of P at $\overline{a} \in \mathbb{F}^m$ denoted $mult(P,\overline{a})$ id the largest integer M such that for every non-negative vector $\overline{i}$ with $wt(\overline{i}) < M$ we have $P^{(\overline{i})}(\overline{a}) = 0$ (If M may be taken arbitrarily large we set $mult(P,\overline{a}) = \infty$)*

Note that $mult(P,\overline{a}) \geq 0$ for all $\overline{a} \in \mathbb{F}^m$.

2.2.1 Basic Properties of Multiplicity

We now translate some of the properties of the Hasse derivative into properties of the multiplicities. We will discuss the properties of multiplicities from [DKSS09]

Proposition 2.2.1. *If $P(\overline{X}) \in \mathbb{F}[\overline{X}]$ and $\overline{a} \in \mathbb{F}^m$ are such that $mult(O,\overline{a}) = n$ then $mult(P^{(\overline{i})},\overline{a}) \geq n - wt(\overline{i})$*

Proof: By assumption, for any $\overline{k}$ with $wt(\overline{k}) < n$, we have $P^{(\overline{k})}(\overline{a}) = 0$. Now take any $\overline{j}$ such that $wt(\overline{j}) < n - wt(\overline{i})$. Using Theorem 2.1.1 (3) we have

$$\left(P^{(\overline{i})}\right)^{(\overline{j})}(\overline{a}) = \binom{\overline{i}+\overline{j}}{\overline{i}} P^{(\overline{i}+\overline{j})}(\overline{a})$$

Since $wt(\overline{i}+\overline{j}) = wt(\overline{i}) + wt(\overline{j}) < m$, hence $\left(P^{(\overline{i})}\right)^{(\overline{j})}(\overline{a}) = 0$. Thus $mult(P^{(\overline{i})},\overline{a}) \geq n - wt(\overline{i})$ ■

We will now discuss the behavior of multiplicites under composition of polynomial tuples. Let $\overline{X} = (X_1, X_2, \ldots, X_m)$ and $\overline{Y} = (Y_1, Y_2, \ldots, Y_n)$ be formal variables. Let $P(\overline{X}) = (P_1(\overline{X}), \ldots, P_k(\overline{X})) \in \mathbb{F}[\overline{X}]^k$ and also $Q(\overline{Y}) = (Q_1(\overline{Y}), \ldots, Q_m(\overline{Y})) \in \mathbb{F}[\overline{Y}]^m$. We define the composition polynomial $P \circ Q(\overline{Y}) \in \mathbb{F}[\overline{Y}]^k$ to be the polynomial $P(Q_1(\overline{Y}), \ldots, Q_m(\overline{Y}))$. In this situation we have the following proposition:

Proposition 2.2.2. *Let $P(\overline{X}), Q(\overline{Y})$ be defined as above. Then for any $\overline{a} \in \mathbb{F}^n$*

$$mult(P \circ Q, \overline{a}) \geq mult(P, Q(\overline{a})) \cdot mult(Q - Q(\overline{a}), \overline{a})$$

In particular, since $mult(Q - Q(\overline{a}), \overline{a}) \geq 1$, we have $mult(P \circ Q, \overline{a}) \geq mult(P, Q(\overline{a}))$

Proof: Let $m_1 = mult(P, Q(\overline{a}))$ and $m_2 = (Q - Q(\overline{a}), \overline{a})$. Clearly $m_2 > 0$. If $m_1 = 0$ the result is obvious. Now assume $m_1 > 0$ (so that $P(Q(\overline{a})) = 0$). Now

$$\begin{aligned}
P(Q(\overline{a}+\overline{Z})) &= P\left(Q(\overline{a}) + \sum_{\overline{i}\neq 0} Q^{(\overline{i})}(\overline{a})\overline{Z}^{\overline{i}}\right) \\
&= P\left(Q(\overline{a}) + \sum_{wt(\overline{i})\geq m_2} Q^{(\overline{i})}(\overline{a})\overline{Z}^{\overline{i}}\right) && [\text{Since } mult(Q-Q(\overline{a}),\overline{a}) = m_2 > 0] \\
&= P(Q(\overline{a}) + h(\overline{Z})) && \left[\text{where } h(\overline{Z}) = \sum_{wt(\overline{i})\geq m_2} Q^{(\overline{i})}(\overline{a})\overline{Z}^{\overline{i}}\right] \\
&= P(Q(\overline{a})) + \sum_{\overline{j}\neq 0} P^{(\overline{j})}(Q(\overline{a}))h(\overline{Z})^{\overline{j}} \\
&= \sum_{wt(\overline{j})\geq m_1} P^{(\overline{j})}(Q(\overline{a}))h(\overline{Z})^{\overline{j}} && [\text{since } mult(P,Q(\overline{a})) = m_1 > 0]
\end{aligned}$$

Since each monomial $\overline{Z}^{\overline{i}}$ appearing in h has $wt(\overline{i}) \geq m_2$ and each occurrence of $h(\overline{Z})$ in $P(Q(\overline{a}+\overline{Z}))$ is raised to the power $\overline{j}$ with $wt(\overline{j}) \geq m_1$ we conclude that $P(Q(\overline{a}+\overline{Z}))$ is of the form $\sum_{wt(\overline{k})\geq m_1\cdot m_2} c_{\overline{k}}\overline{Z}^{\overline{k}}$. This shows that $(P\circ Q)^{(\overline{k})}(\overline{a}) = 0$ for each $\overline{k}$ with $wt(\overline{k}) < m_1 \cdot m_2$. And hence we get the result. ■

Corollary 2.2.3. *Let $P(\overline{X}) \in \mathbb{F}[\overline{X}]$. Let $\overline{a}, \overline{b} \in \mathbb{F}^m$. Let $P_{\overline{a},\overline{b}}(T)$ be the polynomial $P(\overline{a} + T\cdot\overline{b}) \in \mathbb{F}[T]$. Then for any $t \in \mathbb{F}$,*

$$mult(P_{\overline{a},\overline{b}}, t) \geq mult(P, \overline{a} + t\cdot\overline{b})$$

Proof: Let $Q(T) = \overline{a} + T\cdot\overline{b} \in \mathbb{F}[T]^m$. Applying Proposition 2.2.2 and $Q(T)$ we get the desired claim. ■

2.2.2 Strengthening of the Schwartz-Zippel Lemma

Theorem 2.2.4 (Schwartz-Zippel Lemma)**.** *Let $P(\overline{X}) \in \mathbb{F}[\overline{X}]$ be a non-zero polynomial with degree d. Let S be a finite subset of $\mathbb{F}$ with at least d elements in it. If we take $\overline{a} \in S^m$ independently and uniformly at random then*

$$Pr_{\overline{a}\in S^m}[P(\overline{a}) = 0] \leq \frac{d}{|S|}$$

We will prove the strengthening of this lemma using $mult$. Now we need a bound on the number of points that a low-degree polynomial can vanish on with high multiplicity. We state a basic bound on the total number of zeroes (counting multiplicity) that a polynomial can have on a product set S^m.

Theorem 2.2.5 ([DKSS09])**.** *Let $P(\overline{X}) \in \mathbb{F}[\overline{X}]$ be a nonzero polynomial of total degree at most d. Then for any finite $S \subseteq \mathbb{F}$,*

$$\sum_{\overline{a}\in S^m} mult(P, \overline{a}) \leq d\cdot|S|^{m-1}$$

In particular, for any integer $s > 0$

$$Pr_{\overline{a}\in S^m}[mult(P, \overline{a}) \geq s] \leq \frac{d}{s|S|}$$

Proof: We will prove this by induction on m. For the base case when $m = 1$ we will first show that if $mult(P, a) = k$ then $(X - a)^k$ divides $P(X)$. To see this, note that by definition of multiplicity, we have that $P(a + Z) = \sum_i P^{(i)}(a)Z^i$ and $P^{(i)}(a) = 0$ for all $i < k$ we conclude that Z^k divides $P(a + Z)$. And thus $(X - a)^k$ divides $P(X)$. It follows that $\sum_{a \in S} mult(P, a)$ is at most the degree of P.

Now suppose $m > 1$. Let

$$P(\overline{X}) = \sum_{j=0}^{t} P_j(X_1, \ldots, X_{m-1})X_m^j$$

where $0 \leq t \leq d$. Now we have $P_t(X_1, \ldots, X_{m-1}) \neq 0$ and $\deg(P_j) \leq d - j$. For any $a_1, \ldots, a_{m-1} \in S$ let $m_{a_1,\ldots,a_{m-1}} = mult(P_t, (a_1, \ldots, a_{m-1}))$.

Claim 1. *For any* $a_1, \ldots, a_{m-1} \in S$

$$\sum_{a_m \in S} mult(P, \overline{a}) \leq m_{a_1,\ldots,a_{m-1}} \cdot |S| + t$$

Proof: Fix $a_1, \ldots, a_{m-1} \in S$. Let $\bar{i} = (i_1, \ldots, i_{m-1})$ be such that $wt(\bar{i}) = m_{a_1,\ldots,a_{m-1}}$. Since $m_{a_1,\ldots,a_{m-1}} = mult(P_t, (a_1, \ldots, a_{m-1}))$, for all $\bar{j}$ such that $wt(\bar{j}) < m_{a_1,\ldots,a_{m-1}}$, $P_t^{(\bar{j})}(a_1, \ldots, a_{m-1}) = 0$. Hence there exists an $\bar{j}$ such that $wt(\bar{j}) = m_{a_1,\ldots,a_{m-1}}$ and $P_t^{(\bar{j})}(a_1, \ldots, a_{m-1}) \neq 0$. Therefore $P_t^{(\bar{i})}(X_1, \ldots, X_{m-1}) \neq 0$. Letting $(\bar{i}, 0)$ we note that

$$P^{(\bar{i},0)}(X_1, \ldots, X_m) = \sum_{j=0}^{t} P_j^{(\bar{i})}(X_1, \ldots, X_{m-1})X_m^j$$

and therefore $P^{(\bar{i},0)}(X_1, \ldots, X_m)$ is a nonzero polynomial. Now

$$\begin{aligned} mult(P, \overline{a}) &\leq wt(\bar{i}, 0) + mult(P^{(\bar{i},0)}, \overline{a}) && \text{[Proposition 2.2.1]} \\ &\leq m_{a_1,\ldots,a_{m-1}} + mult(P^{(\bar{i},0)}(a_1, \ldots, a_{m-1}, X_m), a_m) && \text{[Corollary 2.2.3]} \end{aligned}$$

Now summing over all $a_n \in S$ and using the $m - 1$ case to $P^{(\bar{i},0)}(a_1, \ldots, a_{m-1}, X_m)$ we have

$$\sum_{a_m \in S} mult(P, \overline{a}) \leq \sum_{a_m \in S} m_{a_1,\ldots,a_{m-1}} + \sum_{a_m \in S} mult(P^{(\bar{i},0)}(a_1, \ldots, a_{m-1}, X_m), a_m) = m_{a_1,\ldots,a_{m-1}} \cdot |S| + t$$

■

Using this result we have

$$\sum_{a_1,\ldots,a_m \in S} mult(P, \overline{a}) \leq \sum_{a_1,\ldots,a_{m-1} \in S} m_{a_1,\ldots,a_{m-1}} + \sum_{a_1,\ldots,a_m \in S} t = \sum_{a_1,\ldots,a_{m-1} \in S} m_{a_1,\ldots,a_{m-1}} + |S|^{m-1}t$$

Now by induction on P_t

$$\sum_{a_1,\ldots,a_{m-1} \in S} m_{a_1,\ldots,a_{m-1}} \leq \deg P_t \cdot |S|^{m-2} \leq (d - t)|S|^{m-1}$$

Hence we get

$$\sum_{a_1,\ldots,a_m \in S} mult(P, \overline{a}) \leq \sum_{a_1,\ldots,a_{m-1} \in S} m_{a_1,\ldots,a_{m-1}} + |S|^{m-1}t \leq (d - t)|S|^{m-1} + t \cdot |S|^{m-1} = d \cdot |S|^{m-1}$$

■

Corollary 2.2.6. *Let $P(\overline{X}) \in \mathbb{F}_q[\overline{X}]$ be a polynomial of total degree at most d. If*

$$\sum_{\overline{a} \in \mathbb{F}_q^m} mult(P, \overline{a}) > d \cdot q^{m-1}$$

then $P(\overline{X}) = 0$

2.3 Multiplicity Code

Definition 2.3.1 (Order s evaluation of P, $P^{(<s)}$)**.** *Let s, d, m be nonnegative integers and let q be a prime power. Let $\Sigma = \mathbb{F}_q^{\binom{m+s-1}{m}} = \mathbb{F}_q^{\{\bar{i}: wt(\bar{i})<s\}}$. For $P \in \mathbb{F}_q[\overline{X}]$ and $\overline{a} \in \mathbb{F}_q^m$ we define the order s evaluation of P at $\overline{a}$, denoted $P^{(<s)}(\overline{a})$, to be the vector $\langle P^{(\bar{i})}(\overline{a}) \rangle_{wt(\bar{i})<s} \in \Sigma$*

Definition 2.3.2 (Multiplicity Code)**.** *The multiplicity code of order s evaluations of degree d polynomials in m variables over $\mathbb{F}_q$ is the code over alphabet Σ and has length q^m (All $P^{(<s)}(\overline{a})$ evaluations at all $\overline{a} \in \mathbb{F}_q^m$). For each polynomial $P \in \mathbb{F}_q[\overline{X}]$ with $\deg P \leq d$ there is a codeword C given by*

$$\mathrm{Enc}_{s,d,m,q}(P) = \langle P^{(<s)}(\overline{a}) \rangle_{\overline{a} \in \mathbb{F}_q^m} \in \Sigma^{q^m}$$

Now we will calculate the rate and distance of multiplicity codes.

Theorem 2.3.1. *Let $\mathcal{C}$ be the multiplicity code of order s evaluations of degree d polynomials in m variables over $\mathbb{F}_q$. Then $\mathcal{C}$ has relative distance $\delta = 1 - \frac{d}{sq}$ and rate $\frac{\binom{d+m}{m}}{\binom{m+s-1}{m} q^m}$*

Proof: The alphabet size equals $q^{\binom{m+s-1}{m}}$. The block-length equals q^m.

To calculate the distance, consider any two codewords $c_1 = \mathrm{Enc}_{s,d,m,q}(P_1)$ and $c_2 = \mathrm{Enc}_{s,d,m,q}(P_2)$ where $P_1 \neq P_2$. For any coordinate $\overline{a} \in \mathbb{F}_q^m$ where the codewords c_1, c_2 agree we have $P_1^{(<s)}(\overline{a}) = P_2^{(<s)}(\overline{a})$. Thud for any such $\overline{a}$, we have $(P_1 - P_2)^{(\boldsymbol{i})}(\boldsymbol{a}) = 0$ for all $\bar{i}$ such that $wt(\bar{i}) < s$. Therefore $mult(P_1 - P_2, \overline{a}) \geq s$. Now using Theorem 2.2.5 the fraction of $\overline{a} \in \mathbb{F}_q^m$ for which $mult(P_1 - P_2, \overline{a}) \geq s$ is at most $\frac{d}{sq}$. Then the minimum distance of the code is at least $1 - \frac{d}{sq}$.

A codeword is specified by giving coefficients to each of the monomials of degree at most d. Thus the number of codewords equals $q^{\binom{d+m}{m}}$. Thus the rate equals

$$\frac{\binom{d+m}{m}}{\binom{m+s-1}{m} q^m} = \frac{\prod_{j=0}^{m-1} (d+m-j)}{\prod_{j=1}^{m} ((s+m-j)q)} \geq \left(\frac{1}{1+\frac{m}{s}}\right)^m \left(\frac{d}{sq}\right)^m \geq \left(1 - \frac{m^2}{s}\right)(1-\delta)^m$$

■

2.4 Local Correction of Multiplicity Codes

Suppose P is a polynomial over $\mathbb{F}_q$ in m variables of degree at most d such that $\Delta(\mathrm{Enc}_{s,d,m,q}(P))$ is small. Let $\overline{a} \in \mathbb{F}_q^m$ where r is the received word. The key idea is to pick many random limes containing $\overline{a}$ and to consider the restriction of r to those lines. With high probability over random direction $\overline{b} \in \mathbb{F}_q^m \setminus \{0\}$ by looking at the restriction of r to the line $\overline{a} + T\overline{b}$ and decoding it we will able to recover the univariate polynomial $P(\overline{a} + T\overline{b})$. This univariate polynomial will tell us a certain linear combination of the various derivatives of P at $\overline{a}$, $\langle P^{(<s)}(\overline{a})\rangle_{wt(\overline{i})<s}$. Combining this for various directions $\overline{b}$, we will know a system of various linear combinations of the numbers $\langle P^{(<s)}(\overline{a})\rangle_{wt(\overline{i})<s}$. Solving this system of linear equations we get $P^{(\overline{i})}(\overline{a})$ for each $\overline{i}$ as desired.

2.4.1 Preliminaries on Restrictions and Derivatives

We first consider the relationship between the derivatives of a multivariate polynomial P and its restrictions to a line. Fix $\overline{a}, \overline{b} \in \mathbb{F}_q^m$ and consider the polynomial $Q(T) = P(\overline{a} + T\overline{b})$

- **The relationship of $Q(T)$ with the derivatives of P at $\overline{a}$:** By the definition of Hasse derivative
$$Q(T) = \sum_{\overline{i}} P^{(\overline{i})}(\overline{a}) b^{\overline{i}} T^{wt(\overline{i})}$$
Then by grouping terms we obtain:
$$\sum_{\overline{i}:wt(\overline{i})=j} P^{(\overline{i})}(\overline{a}) b^{\overline{i}} = \text{Coefficient of } T^j \text{ in } Q(T)$$

- **The relationship of the derivatives of Q at t with the derivatives of P at $\overline{a} + T\overline{b}$:** Let $t \in \mathbb{F}_q$. BY the definition of Hasse Derivatives, we get:
$$P(\overline{a} + \overline{b}(T+R)) = Q(T+R) = \sum_j Q^{(j)}(T) R^j \qquad P(\overline{a} + \overline{b}(T+R)) = \sum_{\overline{i}} P^{(\overline{i})}(\overline{a} + T\overline{b})(\overline{b}R)^{\overline{i}}$$
Therefore comparing the coefficients we obtain:
$$Q^{(j)}(T) = \sum_{\overline{i}:wt(\overline{i})=j} P^{(\overline{i})}(\overline{a} + T\overline{b}) \overline{b}^{\overline{i}}$$

- **The relationship of $Q_{\overline{e}}(T) := P^{(\overline{e})}(\overline{a} + T\overline{b})$ with the derivatives of P at $\overline{a}$:**
$$\sum_{\overline{i}:wt(\overline{i})=j} (P^{(\overline{e})})^{(\overline{i})}(\overline{a}) \overline{b}^{\overline{i}} = \sum_{\overline{i}:wt(\overline{i})=j} \binom{\overline{e}+\overline{i}}{\overline{e}} P^{(\overline{e}+\overline{i})}(\overline{a}) \overline{b}^{\overline{i}} = \text{Coefficient of } T^j \text{ in } Q_{\overline{e}}(T)$$

- **The relationship of the derivatives of $Q_{\overline{w}}$ at T with the derivatives of P at $\overline{a} + T\overline{b}$:** Let $t \in \mathbb{F}_q$.
$$Q_{\overline{e}}^{(j)}(T) = \sum_{\overline{i}:wt(\overline{i})=j} (P^{(\overline{e})})^{(\overline{i})}(\overline{a} + T\overline{b}) \overline{b}^{\overline{i}} = \sum_{\overline{i}:wt(\overline{i})=j} \binom{\overline{e}+\overline{i}}{\overline{e}} P^{(\overline{e}+\overline{i})}(\overline{a} + T\overline{b}) \overline{b}^{\overline{i}}$$

We are now in a position to describe our decoding algorithm. Before describing the main local self-correction algorithm for correcting from $\Omega(\delta)$-fraction errors, we describe a simpler version of the algorithm which corrects from a much smaller fraction of errors.

2.4.2 Simplified Error-Correction from Few Errors

Input: Received word $r : \mathbb{F}_q^m \to \Sigma$, point $\overline{a} \in \mathbb{F}_q^m$.
Output: $P^{(<s)}(\overline{a})$ where $P(\overline{X})$ is such that $\Delta(\mathrm{Enc}_{s,d,m,q}(P), r) \leq \frac{\delta}{100\binom{m+s-1}{m}}$

Abusing notation we will write $r^{(\overline{i})}(\overline{a})$ to mean the $\overline{i}$th coordinate of $r(\overline{a})$.
Algorithm:

1. **Pick a set B of directions:** Choose $B \subseteq \mathbb{F}_q^m \setminus \{0\}$, a uniformly random subset of size $w := \binom{m+s-1}{m}$.

2. **Recover $P(\overline{a} + T\overline{b})$ for directions $\overline{b} \in B$:** For each $\overline{b} \in B$, consider the function $l_{\overline{b}} : \mathbb{F}_q \to \mathbb{F}_q^s$ given by

$$(l_{\overline{b}})_j = \sum_{\overline{i}:wt(\overline{i})=j} r^{(\overline{i})}(\overline{a} + T\overline{b})\overline{b}^{\overline{i}}$$

where $(l_{\overline{b}})_j$ is the jth coordinate of $l_{\overline{b}}(t)$.

Now find the polynomial $Q_{\overline{b}}(T) \in \mathbb{F}[T]$ of degree at most d (if any) such that $\Delta\left(\text{Enc}_{s,d,m,q}(Q_{\overline{b}}), l_{\overline{b}}\right) < \frac{\delta}{2}$

3. **Solve a linear system to recover $P^{(<s)}(\overline{a})$:** For each e with $0 \leq e < s$ consider the following system of equations in the variables $\langle u_{\overline{i}} \rangle_{wt(\overline{i})=e}$ (with one equation for each $\overline{b} \in B$):

$$\sum_{\overline{i}:wt(\overline{i})=e} \overline{b}^{\overline{i}} u_{\overline{i}} = \text{Coefficient of } T^e \text{ in } Q_{\overline{b}}(T)$$

Find all $\langle u_{\overline{i}} \rangle_{wt(\overline{i})=e}$ which satisfy at all these equations. If there are 0 or >1 solutions, output FAIL.

4. Output the vector $\langle u_{\overline{i}} \rangle_{wt(\overline{i})=e}$.

Analysis:

Step 1: All the $\overline{b} \in B$ are "good": For $|obv \in \mathbb{F}_q^m \setminus \{0\}$ we will be interested in the graction of errors on the line $\{\overline{a} + t\overline{b} \mid t \in \mathbb{F}_q \setminus \{0\}\}$ through $\overline{a}$ in direction $\overline{b}$. Since these lines cover $\mathbb{F}_q^m \setminus \overline{a}$ uniformly, we can conclude that at most $\frac{1}{50\binom{m+s-1}{m}} \leq \frac{1}{50} < 0.1$ of the lines containing $\overline{a}$ have more than $\frac{\delta}{2}$ fraction of errors on them. Hence the probability that a line has fewer than $\frac{\delta}{2}$ errors is at least 0.9 over the choice of B.

Step 2: $Q_{\overline{b}}T = P(\overline{a} + T\overline{b})$ for each $\overline{b} \in B$: Assume that B is such that the above event occurs. In this casem for each $\overline{b} \in B$

CHAPTER 3

Expander Codes

The main drawback of Reed-Solomon codes is the large alphabet size. Expander codes are codes that do not have this drawback.

It is a sparse graph with the property that the neighborhood of S (small enough set) in the graph is larger than S itself. To build an error-correcting code, it is best to start with a bipartite expander graph.

3.1 Bipartite Expander Graphs

Definition 3.1.1 ((α, β)-Bipartite Expander Graphs)**.** *A bipartite graph $G = (L, R, E)$ with bipartition L, R is called an (α, β) expander, if for every set $S \subseteq L$ with $|S| \leq \alpha|L|$, the number of vertices in R that are connected to S, i.e. $|\Gamma(S)| \geq \beta|S|$*

We also introduce two notions which we will use wildly:

$$\Gamma^{odd}(S) = \{j \in R \mid |\Gamma(j) \cap S| = \text{odd}\} \quad \Gamma^{+}(S) = \{j \in R \mid |\Gamma(j) \cap S| = 1\}$$

So $\Gamma^{odd}(S)$ is the set of vertices of R which have odd neighbors in S and $\Gamma^{+}(S)$ is the set of unique neighbors of S

Definition 3.1.2 (Unique Neighbor of S)**.** *Given $S \subseteq L$, the vertex $v \in R$ is an unique neighbor of S if v is adjacent to exactly one vertex in S.*

So we get

$$\Gamma^{+}(S) \subseteq \Gamma^{odd}(S) \subseteq \Gamma(S)$$

Remark: For convenience we will take $G = (L, R, E)$ to be (c, d) regular that is G is left c regular and right d regular.

Definition 3.1.3 ((α, β)-Unique Expander)**.** *$G = (L, R, E)$ is a (α, β)-unique expander if $\forall\ S \subseteq L$*

$$|S| \leq \alpha|L| \implies |\Gamma^{+}(S)| \geq \beta|S|$$

Theorem 3.1.1. *If $G = (L, R, E)$ is (α, β)-expander where $\beta > \frac{\alpha}{2}$ then it is a $(\alpha, 2\beta - c)$-unique expander.*

Proof: Take $U = \gamma^+(S)$ and $T = \Gamma(S) \setminus \Gamma^+(S)$. Now we know

$$|U \cup T| = |\Gamma(S)| \geq \beta|S|$$

We will count the number of edges between S and $\Gamma(S)$. Since L is c regular from left side total $c|S|$ edges are there. In right side for unique neighbors there are one edge for each unique neighbor and for other vertices there are at least 2 edges. So from right side at least $|U| + 2|T|$ edges are there. So we have the relation

$$|U| + 2|T| \leq c|S| \implies |U| + 2(|\Gamma(S) - |U|) \leq c|S| \implies |U| \geq 2|\Gamma(S)| - c|S| \geq (2\beta - c)|S|$$

Since we are given that $\beta > \frac{c}{2}$ the graph is $(\alpha, 2\beta - c)$-unique expander. ■

3.2 Expander Code

We will take $|L| = n$ and $|R| = m$ from now on. And also by default we will assume $\beta > \frac{c}{2}$

Definition 3.2.1. *The $m \times n$ adjacency matrix H of the (α, β)-bipartite expander graph. Then H is the parity check matrix of the corresponding expander code C. we denote the corresponding expander code of a (α, β)-expander graph $G = (L, R, E)$ by $\mathcal{C}(G)$.*

Remark: These codes are also called as *Low Density Parity Check Codes*, because the parity check code H is a sparse matrix.

Dimension: Since the parity check matrix is $m \times n$ matrix. The dimension of the code is $n - m = n - \frac{cn}{d} = n\left(1 - \frac{c}{d}\right)$.
Rate: Rate of the code $1 - \frac{c}{d}$.
Distance: The distance of the code is $\geq \alpha n$ (Proved below).

Theorem 3.2.1. *If $G = (L, R, E)$ is a (α, β)-expander with $\beta > \frac{c}{2}$ then*

$$\delta(\mathcal{C}(G)) \geq \alpha$$

Proof: Since the code is linear, it suffices to show that every codeword has hamming weight at least αn. Assume the contrary. Let $c = (c_1, \ldots, c_n) \in \mathcal{C}(G)$ be a nonzero codeword of min weight. Suppose $wt(c) < \delta n$. Take $S = \{i \in L \mid c_i = 1\}$.

Now by Theorem 3.1.1 $|\Gamma^+(S)| \geq (2\beta - c)|S| > 0$. So $\Gamma^+(S)$ is nonempty. Every constraint in $\Gamma^+(S)$ is a violated constraint. Hence contradiction $\lightning$. $wt(c) \geq \alpha n$. ■

Corollary 3.2.2. *$G = (L, R, E)$ is (c, d)-regular $(\delta, c(1 - \epsilon))$-expander for some $\epsilon \in \left(0, \frac{1}{2}\right)$ then $\delta(\mathcal{C}(G)) > \delta$*

From now on we will use the notion of $(\delta, c(1 - \epsilon))$-expander.

Theorem 3.2.3. *$G = (L, R, E)$ is (c, d)-regular $(\delta, c(1 - \epsilon))$-expander for some $\epsilon \in \left(0, \frac{1}{2}\right)$ then*

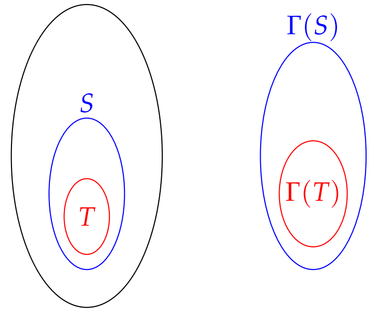
$$\delta(\mathcal{C}(G)) > 2\delta(1 - \epsilon)$$

Proof: Let c is the min weight nonzero codeword. Take $S = \{i \in L \mid c_i = 1\}$. From Theorem 3.2.1 we have $|S| \geq \delta n$. Suppose $|S| < 2\delta(1 - \epsilon)n$ for contradiction. So we have

$$\delta n \leq |S| < 2\delta(1 - \epsilon)n$$

Fix any subset $T \subseteq S$ such that $|T| = \delta n$. Now

$$
\begin{aligned}
|\Gamma^{odd}(S)| &\geq |\Gamma^{+}(S)| \\
&\geq |\Gamma^{+}(T)| - |\Gamma(S \setminus T)| \\
&\geq (c(1-2\epsilon)\delta n) - c|S \setminus T| \\
&> (c(1-2\epsilon)\delta n) - c(\delta(1-2\epsilon))n = 0
\end{aligned}
$$



So $|\Gamma^{odd}(S)| > 0$. Hence there is a vertex $v \in R$ such that there is odd number of neighbors in S. Hence the constraint v is not satisfied. Hence contradiction $\lightning$. ■

3.3 Decoding of Expander Codes

Algorithm 1: Linear Time Decoding Algorithm for Expander Code

Input: $r = (r_1, \ldots, r_n)$ with promise $\exists!\ c \in \mathcal{C}(G)$ such that $\delta(r, c) < \delta(1-2\epsilon)n$

begin

- `// Step 1: (Initialization Phase)`
- $k \longleftarrow 0$
- $x^{(k)} \longleftarrow r$
- **foreach** $j \in R$ **do**
  - **if** $\sum_{i \in \Gamma(j)} x_i = 0$ **then**
    - label j as "SAT"
  - **else**
    - label j as "UNSAT"
- **foreach** $i \in L$ **do**
  - $\mathsf{SAT}_i^{(k)} = \{j \in \Gamma(i) \mid j \text{ labeled "SAT"}\}$
  - $\mathsf{UNSAT}_i^{(k)} = \{j \in \Gamma(i) \mid j \text{ labeled "UNSAT"}\}$
- `// Step 2:`
- **while** $\exists\, i \in L$ *s.t.* $|\mathsf{UNSAT}_i^{(k)}| > |\mathsf{SAT}_i^{(k)}|$ **do**
  - $x_i^{(k+1)} \longleftarrow 1 - x_i^{(k)}$
  - $x_{i'}^{(k+1)} \longleftarrow x_i^{(k)}$ for all $i' \neq i$
  - Update $\mathsf{SAT}_i^{(k)}$ and $\mathsf{UNSAT}_i^{(k)}$
  - $k \longleftarrow k+1$
- `// Step 3:`
- **return** x^k

3.3.1 Analysis

Let r be the received word and $c \in \mathcal{C}(G)$ be the unique codeword such that $\delta(r, c) < \delta(1 - 2\epsilon)n$. Denote

$$S^{(k)} = \{i \in L \mid x_i^{(k)} \neq c_i\}$$

Hence we have $|S^{(0)}| < \delta(1 - 2\epsilon)n$. Also we will use the set $\mathsf{UNSAT}^{(k)}$ to denote the set of unsatisfied right constraints at kth step. Similarly for $\mathsf{SAT}^{(k)}$.

Lemma 3.3.1. *If $\epsilon \in \left(0, \frac{1}{4}\right)$ and $0 < |S^{(k)}| \leq \delta n$ then $\exists\, i \in L$ such that $|\mathsf{UNSAT}_i| > |\mathsf{SAT}_i|$.*

Proof: First notice that all unique neighbors of $S^{(k)}$ are unsatisfied at kth iteration. $\epsilon \in \left(0, \frac{1}{4}\right)$ hence the graph is $(\delta, c(1 - \epsilon))$-expander it is $(\delta, c(1 - 2\epsilon))$-unique expander by Theorem 3.1.1. Hence

$$|\mathsf{UNSAT}^{(k)}| \geq c(1 - 2\epsilon)|S^{(k)}| > \frac{c}{2}|S^{(k)}|$$

Hence, $\exists\, i \in S^{(k)}$ such that $|\mathsf{UNSAT}_i^{(k)}| > \frac{c}{2}$. Now the degree of i is c. Hence $|\mathsf{UNSAT}_i^{(k)}| > |\mathsf{SAT}_i^{(k)}|$. ■

Since for each iteration the distance between $x^{(k)}$ and c is at most δn, c is the only codeword which is nearest to $x^{(k)}$. Hence the nearest codeword for each iteration stays the same.

Now in the algorithm there are two things to observe.

Observation 1. *The number of unsatisfied right constraints is always decreasing.*

Observation 2. $|S^{(k)} - S^{(k+1)}| = 1$

Lemma 3.3.2. $|S^{(0)}| < \delta(1 - 2\epsilon)n \implies |S^{(k)}| < \delta n.$

Proof: Initially $\mathsf{UNSAT}^{(0)} \subseteq \Gamma(S^{(0)})$ since the unsatisfied constraints are the subset of the neighbors of errors. Hence

$$|\mathsf{UNSAT}^{(0)}| \leq |\Gamma(S^{(0)})| \leq c|S^{(0)}| < c|S^{(o)}| < c\delta(1 - 2\epsilon)n$$

Suppose there exists a k' such that $|S^{(k')}| \geq \delta n$. By the observation there exists k such that $|S^{(k)}| = \delta n$. Hence

$$|\mathsf{UNSAT}^{(k)}| > |\Gamma^{+}(S^{(k)})| \geq \delta n \cdot c(1 - 2\epsilon)$$

But the $|\mathsf{UNSAT}^{(k)}|$ keeps decreasing so it can not start with less than $c\delta(1 - 2\epsilon)n$ and at some point is $\geq \delta n \cdot c(1 - 2\epsilon)$. Hence contradiction ↯ . ■

At kth iteration suppose the number of unsatisfied constraints is nonzero and $|S^{(k)}| < \delta n$. Since number of unsatisfied constraints is nonzero $|S^{(k)}| > 0$. By Lemma 3.3.1 there exists an $i \in L$ such that $|\mathsf{UNSAT}_i| > |\mathsf{SAT}_i|$. Hence the algorithm will find some vertex which has more unsatisfied constraints than satisfied constraints and flip its bit and proceed to the next iteration. With this process the number of unsatisfied constraints reduced by at least 1. Thus the algorithm will keep reducing the number of unsatisfied constraints till it becomes zero because if its not zero at any jth iteration and then $|S^{(j)}| > 0$ and hence by the above argument it will proceed. Once the number of unsatisfied constraints becomes zero cause then the final output, suppose x satisfies all the right constraints. Hence it is indeed a codeword and since the nearest codeword at each iteration stays the same $x = c$.

3.3.2 Time Complexity

1. Preprocessing Stage: For each $j \in R$ to check $\sum_{i \in \Gamma(j)} x_i = 0$ it takes $O(d)$ time. Hence the first for loop takes $O(md)$ time. Now for each vertex in L we keep the number of unsatisfied constraints which are neighbor of that vertex. We also keep a list of vertices in L which have more unsatisfied constraints than satisfied constraints. This can be done in $O(cn)$ time.

2. In each iteration of the while loop instead of searching for a vertex with more unsatisfied constraints than satisfied constraints we remove an element of Q.

   After flipping the vertex we update the list of unsatisfied constraints in R in $O(c)$ time. Then we will update the number of unsatisfied constraints associated with each element of in L which are neighbors of the neighbors of i i.e. the vertices in $\Gamma(\Gamma(i))$ in $O(cd)$ time. Since after the bit flip the previously unsatisfied constraints are satisfied in $\Gamma(i)$ and the previously satisfied constraints are now unsatisfied. For each vertex $j \in \Gamma(i)$ if j was previously unsatisfied then we will subtract 1 from the number of unsatisfied constrains of the neighbors of j and if j was previously satisfied then we will add 1 for any previously satisfied constraint to the number of unsatisfied constrains of the neighbors of j. Now from Q we will remove the elements which have lesser unsatisfied constraints than satisfied constraints and add the.

   After updating the number of unsatisfied constraints of each vertex in $\Gamma(\Gamma(i))$ we will add the vertices which have more unsatisfied constraints than satisfied constraints into Q and remove the vertices which have lesser unsatisfied constrains than satisfied constraints. This all can be done in $O(cd)$ time since $|\Gamma(\Gamma(i))| \leq cd$. Since c, d are constants every thing inside each iteration can be done in constant time.

3. In each iteration the number of unsatisfied constraints reduces by at least 1. The original number of unsatisfied constraints is at most $c\delta(1-2\epsilon)n$. (Theorem 3.3.2). Then the total number of iterations is at most $c\delta(1-2\epsilon)n = O(n)$.

Hence the algorithm decodes the received word in $O(n)$ time.

CHAPTER 4

Appendix

4.1 Probabilistic Inequalities

Theorem 4.1.1 (Markov's Inequality)**.** *For any random variable X and $a > 0$*

$$Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$$

Theorem 4.1.2 (Chebyshev's Inequality)**.** *For a random variable with variance σ, and expected value μ with $a \geq 0$*

$$Pr[|X - \mu| \geq a] \leq \frac{Var[X]}{a^2}$$

CHAPTER 5

Bibliography

[DKSS09] Zeev Dvir, Swastik Kopparty, Shubhangi Saraf, and Madhu Sudan. Extensions to the method of multiplicities, with applications to kakeya sets and mergers. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 181–190, 2009.

[Kop15] Swastik Kopparty. Some remarks on multiplicity codes. *CoRR*, abs/1505.07547, 2015.

[KSY11] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. volume 61, pages 167–176, 06 2011.

[Yek12] Sergey Yekhanin. Locally decodable codes. *Foundations and Trends in Theoretical Computer Science*, 6(3):139–255, 2012.