# 1 Improvement to List Decoding Algorithm

Last time we looked at the list decoding algorithm for Reed Solomon codes with $1 - \sqrt{2R}$ fraction of errors. Now we will improve this algorithm to list decode up to $1 - \sqrt{R}$ fraction of errors which is the Johnson radius for rate $R$. We begin by recalling the previous algorithm due to Sudan[Sud97].

## 1.1 Previous Algorithm

Given a field $\mathbb{F}$ such that $|\mathbb{F}| > n$ and $n$ points $(a_i, y_i) \in \mathbb{F}^2$, we want to find all polynomials $f(x) \in \mathbb{F}[x]$ with degree at most $k$ such that $f(a_i) = y_i$ for at least $t \geq n - e$ values of $1 \leq i \leq n$. Here $e$ is the number of errors so $t$ is the number of points of agreement.

Step 1. Fit a nonzero polynomial $Q(x, y) = A_0(x) + A_1(x) + \cdots + A_L(x)y^L$ of degree $D$ with $\deg A_j \leq D - kj$ such that $Q(a_i, y_i) = 0, i = 1, \ldots, n$.

Step 2. Find all $f(x)$ such that $y - f(x) \mid Q(x, y)$ and output those with degree at most $k$ and agreement at least $t$.

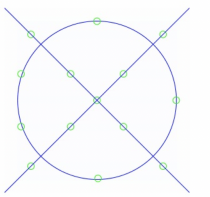
Then our analysis relied on recognizing that $y - f(x) \mid Q(x, y) \Leftrightarrow R_f(x) := Q(x, f(x)) = 0$ so $\deg R_f \leq D$. On the other hand, $R_f(a_i) = 0$ whenever $f(a_i) = y_i$. Hence, if $t > D$ then $R_f \equiv 0$. We showed $D \geq \frac{n}{L+1} + \frac{kL}{2}$ was sufficient by counting the number of coefficients of $Q(x, y)$ and requiring it to be larger than the number of constraints.

## 1.2 Intuition

The geometric intution we had for the above algorithm was the following.

**Example 17.1.** Consider the following point configuration on which we want to find all the lines passing through at least 5 points.

By inspecting the figure, we can see that there are only two such lines $f_1(x) = x$ and $f_2(x) = -x$:



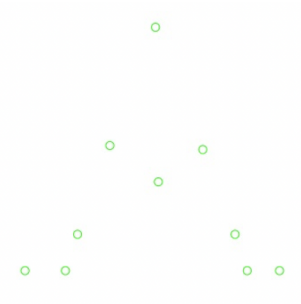
**Claim 17.2.** *Any total degree 4 polynomial $Q(x, y)$ such that $Q(\alpha_i, y_i) = 0$ for all $i$, must have these two lines as factors.*

*Proof.* Both lines must intersect $Q(x, y) = 0$ in at lesat 5 points. Let $Q(\alpha_j, f_i(\alpha_j)) = 0$ for five different points. Since $Q(x, f_i(x))$ is a degree$-4$ polynomial in $x$, we have $Q(x, f_i(X)) \equiv 0$. Thus, $y - f_i(x) \mid Q(x, y)$.

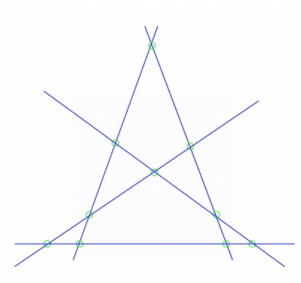
## 2 Method of Multiplicities

In this section, we will remove the factor $\sqrt{2}$ and obtain a list decoding algorithm with $t > \sqrt{kn}$ due to Guruswami and Sudan[GS98]. First we will see an example showing that we can't hope to improve the above parameters.

**Example 17.3.** Consider the following point configuration of $n = 10$ points. For $k = 1$ (lines), having $t \geq \sqrt{nk}$ implies $t > 3$. So we want to find lines passing through at least 4 points.



Next figure shows set of all lines going through at least 4 points:



If we want to output all these 5 lines, $Q$ must have total degree at least 5. But the agreement parameter $t = 4$ is smaller than 5, so $Q(x, f(x))$ (for one of these lines; $y = f(x)$) might not be $\equiv 0$. Note that in this example, each point has two lines crossing. Thus any $Q(x, y)$ containing these lines as factors must also contain each point twice.

Now we will try to understand what it means for Q(x, y) to contain each point twice.

**Example 17.4.** If $Q(x, y) \in \mathbb{F}[X, Y]$ has $w$ zereos at point $(0, 0)$, then it has no monomials of total weight less than $w$.

Since we can translate $Q$ by any $(\alpha, \beta)\mathbb{F}^2$, we can generalize the above example to a formal definition:

**Definition 17.5.** A polynomial $Q(X, Y)$ is said to have a zero of multiplicity $w \geq 1$ at a point $(\alpha, \beta) \in \mathbb{F}^2$ if $Q_{\alpha,\beta} := Q(x + \alpha, y + \beta)$ has no monomials of total weight less than $w$ i.e. no $x^i y^j$ term with $i + j < w$.

Armed with this definition, we update our earlier algorithm to the following.

Step 1. Fit a nonzero polynomial $Q(x,y) = A_0(x) + A_1(x) + \cdots + A_L(x)y^L$ of degree $D$ with $\deg A_j \leq D - kj$ such that $Q(a_i, y_i)$ has a zero of weight $w$, $i = 1, \ldots, n$.

Step 2. Find all $f(x)$ such that $y - f(x) \mid Q(x,y)$ and output those with degree at most $k$ and agreement at least $t$.

Notice that for each $(\alpha, \beta)$ such that $Q(\alpha, \beta) = 0$ with weight $w$, we obtain $\binom{w+1}{2}$ homogenous linear equations in coefficients of $Q(x,y)$. Hence, we get $n\binom{w+1}{2}$ total linear equations.

Now we present a couple lemmas that will finish off the argument.

**Lemma 17.6.** *Suppose $f(\alpha) = \beta$ and $Q(x,y)$ has $w$ zeroes at $(\alpha, \beta)$ then $(x-\alpha)^w \mid Q(x, f(x))$.*

*Proof.* We have

$$R_f(x) = Q(x, f(x)) = Q_{\alpha,\beta}(x - \alpha, f(x) - \beta) = Q_{\alpha,\beta}(x - \alpha, f(x) - f(\alpha).$$

$Q_{\alpha,\beta}$ has no monomials of degree $< w$. Also $x - \alpha \mid f(x) - f(\alpha)$. For any nonzero monomial $x^i y^j$ of $Q_{\alpha,\beta}$, we have $(x-\alpha)^{i+j} \mid (x-\alpha)^i (f(x) - f(\alpha))^j$ and since $i + j \geq w$, we have

$$(x-\alpha)^w \mid Q_{\alpha,\beta}(x - \alpha, f(x) - \beta) = Q(x, f(x)).$$

In order to have a solution the the linear equations, we need

$$\begin{aligned}(D+1)(L+1) - kL(L+1)/2 &> n\binom{w+1}{2} \\ \implies D + 1 &> \frac{nw(w+1)}{2(L+1)} + \frac{kL}{2} \\ \implies D &\geq \frac{nw(w+1)}{2(L+1)} + \frac{kL}{2}\end{aligned}$$

so taking

$$D \geq \frac{nw(w+1)}{2L} + \frac{kL}{2}$$

suffices. We also obtain the following corollary.

**Corollary 17.7.** *If a degree $k$ polynomial $f$ has $f(\alpha_i) = y_i$ for $\geq t$ values of $i$ and $wt > D$ then $R_f(x) = 0$.*

*Proof.* By the previous lemma we have

$$\prod_{i : f(\alpha_i) = y_i} (x - \alpha_i)^w \mid R_f(x)$$

so if $wt > D$ then we must have $R_f(x) \equiv 0$.

Additionally, for $wt > D$ we need

$$t > \frac{n(w+1)}{2L} + \frac{kL}{2w}$$

so taking

$$L \approx \sqrt{\frac{nw(w+1)}{k}}$$

we get

$$t > \sqrt{\frac{nk(w+1)}{w}} = \sqrt{nk + \frac{1}{w}}.$$

Thus, we get the following corollary.

**Corollary 17.8.** *For a Reed Solomon code of rate $R$, we can list decode up to $1 - \sqrt{R}$ fraction of errors in time polynomial in the block length and field size.*

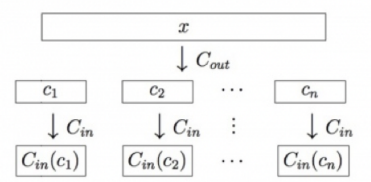
*Proof.* Exercise.

# 3 List Decoding of Binary Concatenated Codes

## 3.1 Construction

The main shortcoming of the above methods is that the alphabet size is polynomial in block length. We will use concatenated codes to reduce alphabet size.

Assume we are given a Reed-Solomon code as out outer code $C_{\text{out}} : \mathbb{F}_2^n \to \mathbb{F}_2^{n/R}$ with rate $R$ and an inner code $C_{\text{in}} : \mathbb{F}_2^n \to \mathbb{F}_2^{m/r}$ with rate $r = \text{rate}(C_{\text{in}})$. Consider the basic composition:




This code has rate $R_0 = R/r$. First we will start with the most naive idea and work our way up. Our main goal in this section is to get a binary code which is close to the Zyablov bound.

### 3.1.1 Uniquely Decoding Inner Codes

Consider uniquely decoding each inner code and then applying the list decoding algorithm on the outer code. There will be $\geq \frac{h^{-1}(1-r)}{2}$ fraction of errors if an inner block is decoded to a wrong codeword. The list decoding capacity of $C_{\text{out}}$ is $1 - \sqrt{\frac{R}{r}} = 1 - \sqrt{R_0}$. Hence this algorithm can list decode up to $(1 - \sqrt{R_0})\frac{h^{-1}(1-r)}{2}$ fraction of errors. However, this is quite bad, as we picked up a factor of $1/2$ that we were trying to avoid by using list decoding at the first place.

**Lemma 17.9.** *Given positive reals $0 < r, R < 1$, there exists a binary code which can be list decoded up to $\left(1 - \sqrt{R/r}\right)\frac{h^{-1}(1-r)}{2}$ fraction of errors in polynomial time.*

*Proof.* Given.

## 3.2 List Decoding Inner Codes

Instead of uniquely decoding inner codes, consider list decoding them, by e.g. brute force. Recall that by the Johnson Bound, any binary code can be list decoded up to $h^{-1}(1 - r - \epsilon)$ fraction of errors with a list size of $\ell = \ell(\epsilon) \leq O(1/\epsilon)$.

A slight complication arises on how to apply RS list decoding on a message where each symbol is another list. During the list decoding algorithm, we only assumed all $(\alpha_i, y_i$ pairs were distinct, and everything worked fine even in $\alpha_i = \alpha_j$ for different $i, j$ as long as $y_i \neq y_j$. Using this observation, we can apply our list decoding algorithm on the set $\{(\alpha_i, y_{ij})\}_{i,j}$. This will return us a list of messages which agrees with $t > \sqrt{kN}$ points where $N < n\ell$. Therefore, this method will give us list decoding up to $\left(1 - \sqrt{R/r}\right) h^{-1}(1 - r - \epsilon)$ fraction of errors.

**Lemma 17.10.** *Given positive rate $0 < r, R < 1$ and for any $0 < \epsilon < r$, there exists a binary linear code which can be list decoded up to $\left(1 - \sqrt{R/r}\right) h^{-1}(1 - r - \epsilon)$ fraction of errors.*

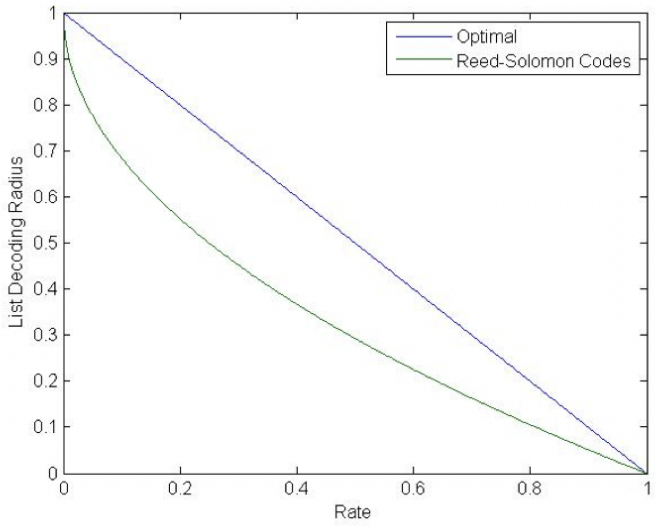
In order to get a binary code list decodable up to $(1-\gamma)/2$ fraction of errors, we can take $\ell = 1/\gamma^2$ and $r = O(\gamma^2)$, which implies $\epsilon = O(\gamma^2)$. By taking $R = r^5$, we obtain:

**Theorem 17.11.** *For any real $0 < \gamma < 1$, there exists a linear binary code of rate $\Omega(\gamma^6)$ list decodable up to $(1-\gamma)/2$ fraction of errors with a list size of $O(1/\gamma^3)$*

**Remark 17.12.** Compare this to the optimal paramter of rate $\Omega(\gamma^2)$ and list size $O(1/\gamma^2)$.

# 4 Going Beyond Reed-Solomon Codes & Johnson Radius

For Reed-Solomon codes, we showed that one can efficiently list decode up to a radius (fraction of errors) equal to $1 - \sqrt{R}$. However, we know that list decoding up to a fraction $1 - R - \epsilon$ of errors is possible non-constructively. In this section, we will construct better variants of Reed-Solomon codes. The code we will describe was given by Guruswami and Rudra [GR08].




The main difficulty in Reed-Solomon codes is that we need to be able to deal with any patter of $e$ errors. Instead, we will "pack" parts of a codeword together and decrease the error patterns we have to handle considerably.

## 4.1 Folded Reed Solomon Codes

Recall that a Reed Solomon code $\mathrm{RS}_{\mathbb{F},\mathbb{F}^*}[n,k]$ gives an encoding of a degree $k$ polynomial $f(x) \in \mathbb{F}[X]$ as

$$f(x) \mapsto (f(\alpha)|_{\alpha \in \mathbb{F}^*}) = (f(1), f(\gamma), \ldots, f(\gamma^{n-1}))$$

where $n = |\mathbb{F}| - 1 = q - 1$ and $\gamma$ is a generator of $\mathbb{F}$.

Folded Reed Solomon code $\mathrm{FRS}^{(s)}_{\mathbb{F}}[k]$ (a code over $\mathbb{F}^*$) is the $s$-folded version of the above (for

convenience assume $s \mid n$):

$$f(x) \mapsto \left( \begin{bmatrix} f(1) \\ f(\gamma) \\ \vdots \\ f(\gamma^{s-1}) \end{bmatrix}, \begin{bmatrix} f(\gamma^{s}) \\ f(\gamma^{s+1}) \\ \vdots \\ f(\gamma^{2s-1}) \end{bmatrix}, \ldots, \begin{bmatrix} f(\gamma^{n-s}) \\ f(\gamma^{n-s+1}) \\ \vdots \\ f(\gamma^{n-1}) \end{bmatrix} \right).$$

The received word looks like following:

| | $1$ | $\gamma^s$ | $\gamma^{2s}$ | $\cdots$ | $\gamma^{n-s}$ |
|---|---|---|---|---|---|
| $1$ | $y_1$ | $y_{s+1}$ | $y_{2s+1}$ | $\cdots$ | $y_{n-s+1}$ |
| $\gamma$ | $y_2$ | $y_{s+2}$ | $y_{2s+2}$ | $\cdots$ | $y_{n-s+2}$ |
| $\gamma^2$ | $y_3$ | $y_{s+3}$ | $y_{2s+3}$ | $\cdots$ | $y_{n-s+3}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ |
| $\gamma^{s-1}$ | $y_s$ | $y_{2s}$ | $y_{3s}$ | $\cdots$ | $y_n$ |

If we think of Reed-Solomon decoding as interpolating a polynomial over a plane (which led to the $\sqrt{R}$ bound on agreement required), it might seem possible to decode with agreement fraction about $R^{s/(s+1)}$ by interpolating $(s+1)$ dimensions.

**Problem 17.13.** We want to find a polynomial $Q(X, Y_1, Y_2, \ldots, Y_s) \not\equiv 0 \in \mathbb{F}[X, Y_1, Y_2, \ldots, Y_s]$ such that $Q(\gamma^{is}, y_{is+1}, y_{is+2}, \ldots, y_{(i+1)s}) = 0$ for $0 \leq i < n/s$.

## References


[GR08] Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on information theory*, 54(1):135–150, 2008. 4

[GS98] Venkatesan Guruswami and Madhu Sudan. Improved decoding of reed-solomon and algebraic-geometric codes. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*, pages 28–37. IEEE, 1998. 2

[Sud97] Madhu Sudan. Decoding of reed solomon codes beyond the error-correction bound. *Journal of complexity*, 13(1):180–193, 1997. 1