Error Correcting Codes: Combinatorics, Algorithms and Applications **(Fall 2007)**

Lecture 39: GS Decoder Wrap-up

November 30,2007

Lecturer: Atri Rudra *Scribe: Kanke Gao*

In the last lecture, we introduced the Guruswami-Sudan (GS) list decoding algorithm for RS codes. We restate the algorithm here. Recall that the input is the received word, in the form $(\alpha_i, y_i) \in \mathbb{F}^2$, for $1 \le i \le n$.

GS list decoding algorithm

Inputs: $(\alpha_i, y_i) \in \mathbb{F}^2$ for $1 \le i \le n$, agreement parameter $0 \le t \le n$

Step 1) Compute a non-zero $Q(X, Y)$, such that
i) $(1, k)$ weighted degree of $Q \le D$.
ii) Q has r roots at (α_i, y_i), $1 \le i \le n$.
Step 2) Output all degree$\le k$ polynomials $P(X)$ such that
i) $Y - P(X)$ divides $Q(X, Y)$
ii) $P(\alpha_i) = y_i$ for at least t positions i.

We analyzed the GS list decoding algorithm modulo two lemmas. In today's lecture, we are going to prove the two lemmas.

1 Proof of key lemmas

We now recall the two lemmas:

Lemma 1.1. *The condition that $Q(X, Y)$ has r roots at (α, β) implies $\binom{r+1}{2}$ constraints on the coefficients of Q.*

Lemma 1.2. *If $Q(X, Y)$ is output by **Step 1** and $P(X)$ needs to be output in **Step 2**, then $Y - P(X)$ divides $Q(X, Y)$.*

Proof of Lemma 1.1. Let
$$Q(X, Y) = \sum_{\substack{i,j \\ i+kj \le D}} q_{i,j} X^i Y^j$$
and $Q_{\alpha,\beta}(X, Y) = Q(X + \alpha, Y + \beta) = \sum_{i,j} q^{\alpha,\beta}_{i,j} X^i Y^j$. We will show that

(i) $q_{i,j}^{\alpha,\beta}$ are homogeneous linear combinations of $q_{i,j}$'s.

(ii) If $Q_{\alpha,\beta}(X,Y)$ has no monomial of degree $< r$, then that implies $\binom{r+1}{2}$ constraints on $q_{i,j}^{\alpha,\beta}$'s.

Note that (i) and (ii) prove the lemma. To prove (i), note that by the definition:

$$Q_{\alpha,\beta}(X,Y) = \sum_{i,j} q_{i,j}^{\alpha,\beta} X^i Y^j \tag{1}$$

$$= \sum_{\substack{i',j' \\ i'+kj' \leq D}} q_{i',j'}(X+\alpha)^{i'}(Y+\beta)^{j'} \tag{2}$$

Note that, if $i > i'$ or $j > j'$, then $q_{i,j}^{\alpha,\beta}$ doesn't depend on $q^{i',j'}$. By comparing coefficients of X^iY^j from (1) and (2), we obtain

$$q_{i,j}^{\alpha,\beta} = \sum_{\substack{i'>i \\ j'>j}} q_{i',j'} \binom{i'}{i}\binom{j'}{j}\alpha^i\beta^j,$$

which proves (i). To prove (ii), recall that by definition $Q_{\alpha,\beta}(X,Y)$ has no monomial of degree $< r$. In other words, we need to have constraints $q_{i,j}^{\alpha,\beta} = 0$ if $i+j \leq r-1$. The number of such constraints is

$$|\{(i,j)|i+j \leq r-1,\ i,j \in \mathbb{Z}^{\geq 0}\}| = \binom{r+1}{2},$$

where the equality follows from the argument we used to bound the dimension of Reed-Muller codes. □

We now re-state Lemma 1.2 more precisely and then prove it.

Lemma 1.3. *Let $Q(X,Y)$ be computed by **Step 1**. Let $P(X)$ be a polynomial of degree $\leq k$, such that $P(\alpha_i) = y_i$ for at least $t > \frac{D}{r}$ many values of i, then $Y - P(X)$ divides $Q(X,Y)$.*

Proof. Define

$$R(X) \triangleq Q(X, P(X)).$$

As usual, to prove the lemma, we will show that $R(X) \equiv 0$. To do this, we will need the following claim.

Claim 1.4. *If $P(\alpha_i) = y_i$, then $(X-\alpha_i)^r$ divides $R(X)$, that is α_i is a root of $R(X)$ with multiplicity r.*

Note that by definition of $Q(X,Y)$ and $P(X)$, $R(X)$ has degree $\leq D$. Assuming the above claim is correct, $R(X)$ has at least tr roots. Therefore, $R(X)$ is a zero polynomial as $tr > D$. We will now prove Claim 1.4. Define

$$P_{\alpha_i,y_i}(X) \triangleq P(X+\alpha_i) - y_i, \tag{3}$$

and

$$\begin{aligned}
R_{\alpha_i,y_i}(X) &\stackrel{\triangle}{=} R(X+\alpha_i) && (4)\\
&= Q(X+\alpha_i, P(X+\alpha_i)) && (5)\\
&= Q(X+\alpha_i, P_{\alpha_i,y_i}(X)+y_i) && (6)\\
&= Q_{\alpha_i,y_i}(X, P_{\alpha_i,y_i}(X)), && (7)
\end{aligned}$$

where the second, third and fourth equalities follow from the definitions of $R(X)$, $P_{\alpha_i,y_i}(X)$ and $Q_{\alpha_i,y_i}(X,Y)$ respectively.

By (4) if $R_{\alpha_i,y_i}(0)=0$, then $R(\alpha_i)=0$. So if X divides $R_{\alpha_i,y_i}(X)$, then $X-\alpha_i$ divides $R(X)$. Similarly, if X^r divides $R_{\alpha_i,y_i}(X)$, then $(X-\alpha_i)^r$ divides $R(X)$. Thus, to prove the lemma, we will show that X^r divides $R_{\alpha_i,y_i}(X)$. Since $P(\alpha_i)=y_i$ when α_i agrees with y_i, we have $P_{\alpha_i,y_i}(0)=0$. Therefore, X is a root of $P_{\alpha_i,y_i}(X)$, that is, $P_{\alpha_i,y_i}(X)=X\cdot g(X)$ for some polynomial $g(X)$ of degree at most $k-1$. We can rewrite $R_{\alpha_i,y_i}(X)=\sum_{i',j'} q_{i',j'}^{\alpha_i,y_i} X^{i'}(P_{\alpha_i,y_i}(X))^{j'} = \sum_{i',j'} q_{i',j'}^{\alpha_i,y_i} X^{i'}(Xg(X))^{j'}$. Now for every i',j' such that $q_{i',j'}^{\alpha_i,y_i}\neq 0$ $i'+j'\geq r$ as $Q_{\alpha_i,y_i}(X,Y)$ has no monomial of degree $<r$. Thus $R_{\alpha_i,y_i}(x)$ has no non-zero monomial X^ℓ, $\ell<r$. Thus X^r divides $R_{\alpha_i,y_i}(X)$, as desired.

□

From the second property of **Step 1**, $Q(X,Y)$ has $r\geq 0$ roots at (α_i,y_i), $1\leq i\leq n$. However, our analysis did not explicitly use the fact that the multiplicity is same for every i. In particular, given non-zero integer multiplicities $w_i\geq 0$, $1\leq i\leq n$, the GS algorithm can output all polynomials $P(X)$ of degree at most k, such that

$$\sum_{i:P(\alpha_i)=y_i} w_i > \sqrt{kn\sum_{i=0}^{n}\binom{w_i+1}{2}}$$

Note that till now we have seen the special case $w_i=r$, $1\leq i\leq n$. Further, note that the α_i's need not be distinct for the all of the previous arguments to go through. In particular, we can further generalize the input which now has positive integer weights $w_{i,\alpha}$ for every $1\leq i\leq n$ and $\alpha\in\mathbb{F}$ and the GS algorithm will be able to output all $P(X)$ of degree at most k such that

$$\sum_i w_{i,P(\alpha_i)} > \sqrt{kn\sum_{i=0}^{n}\sum_{\alpha\in\mathbb{F}}\binom{w_{i,\alpha}+1}{2}}.$$

This will be useful to solve the following generalization of list decoding called soft decoding.

Definition 1.5. *Under soft decoding problem, the decoder is given as input a set of non-negative weights $w_{i,d}(1\leq i\leq n,\alpha\in\mathbb{F}_q)$ and a threshold $W\geq 0$. The soft decoder needs to output all codewords $(c_1,c_2,\ldots,c_n)$ in q-ary code of block length n that satisfy:*

$$\sum_{i=1}^{n} w_{i,c_i}\geq W.$$

Consider the following special case of soft decoding where $w_{i,y_i} = 1$ and $w_{i,\alpha} = 0$ for $\alpha \in \mathbb{F} \setminus \{y_i\}$ $(1 \le i \le n)$. Note that this is exactly the list decoding problem with the received word $(y_1, \ldots, y_n)$. Thus, list decoding is indeed a special case of soft decoding. Soft decoding has practical application in setting where the channel is analog. In such a situation, the "quantizer" might not be able to pinpoint a received symbol y_i with 100% accuracy. Instead it can use the weight $w_{i,\alpha}$ to denote it confidence level that ith received symbol was α. Soft decoding (and its special case list recovery, which we will study in the next lecture) also has application in designing list decoding algorithm for concatenated codes.