Information and Coding Theory **Autumn 2017**

Lecture 13: November 20, 2017

Lecturer: Madhur Tulsiani

1 List-decoding of Reed-Solomon codes

The decoding algorithm in the previous lecture requires the number of errors to be at most $\lfloor \frac{n-k}{2} \rfloor$, i.e. it requires error rate to be less than roughly $\frac{1}{2}(1 - \frac{k}{n}) \approx \frac{1}{2}$. Of course $1/2$ is a bound on the error rate (in the Hamming model) for *any code*, since the number of errors can be at most half the distance.

The notion of list-decoding allows us to toterate more errors, at the cost of producing a (short) list of multiple codewords when it is not possible to decide on a unique closest codeword. We will describe the algorithm by Sudan [Sud97], which list-decodes Reed-Solomon codes up to error rate $1 - 2\sqrt{k/n}$. For an detailed discussion of several results on list decoding, see the excellent survey by Guruswami [Gur07].

We can view the list decoding algorithm below as a generalization of the unique decoding algorithm discussed in the previous lecture. For unique decoding (from t errors), we found polynomials Q and E with degrees $k - 1 + t$ and t respectively, such that

$$y_i \cdot E(a_i) \;=\; Q(a_i) \qquad \forall i \in [n]\,,$$

where $a_1, \ldots, a_n$ are the evaluation points defining the code, and $y_1, \ldots, y_n$ are the (possibly corrupted) received values. This can be seen as finding a curve $R(x, y)$ with $\deg_y(R) = 1$, which passes through the points (a_i, y_i) for all $i \in [n]$. For $R(x, y) = y \cdot E(x) - Q(x)$, we proved that $y - P(x)$ must be a factor of $R(x, y)$, where $P(x)$ is the polynomial defining the closest codeword.

In the case of list decoding, we still find a polynomial $R(x, y)$passing through all the points (a_i, y_i) but allow a larger degree for y. We will show that for any polynomial P in the desired error radius, $y - P(x)$ must be a factor of $R(x, y)$. We define the algorithm below, in terms degree parameters d_x and d_y to be chosen later. Instead of thinking about the number of corruptions t (which will be large here) we will consider the number of agreements $r = n - t$. Also, note that the algorithm requires computing all factors of $R(x, y)$ of the form $y - f(x)$. This can be done efficiently (in time $\mathrm{poly}(p)$) though we do not discuss the details here. See Guruswami's survey for details of this step [Gur07].

List-decoding for Reed-Solomon codes

Input: $\{(a_i, y_i)\}_{i=1,\ldots,n}$

Parameters: $d_x, d_y, r \in \mathbb{N}$

1. Find nonzero $R \in \mathbb{F}_p[x,y]$ such that $\deg_x(R) \leq d_x - 1$, $\deg_y(R) \leq d_y - 1$ and $R(a_i, y_i) = 0$ for each $i \in [n]$.
2. Compute all factors of R that are of the form $y - f(x)$.
3. Output all f from Step 2 such that $|\{i \in [n] \mid f(a_i) = y_i\}| \geq r$.

Lemma 1.1 *There exists $Q(x,y)$ that satisfies the conditions in Step 1 of the algorithm, if d_x, d_y satisfy $d_x \cdot d_y > n$.*

Proof: We observe that finding R is again equivalent to solving linear system. By writing $R(x,y) = \sum_{0 \leq i < d_x} \sum_{0 \leq j < d_y} c_{i,j} x^i y^j$, the equation $R(a_i, y_i) = 0$ for $i \in [n]$ gives n linear equations in the coefficients $c_{i,j}$'s. Note that there are $d_x \cdot d_y$ unknowns and n equations. Also, $c_{i,j} = 0$ for all i, j is a solution, since the system is homogeneous. Thus, if $d_x \cdot d_y > n$, there exist multiple solutions and one of them must be nonzero. ■

Lemma 1.2 *Let $R \in \mathbb{F}_p[x,y]$ that satisfies the conditions in Step 1 of the algorithm. Let $P \in \mathbb{F}_p[x]$ be a polynomial with degree at most $k-1$, such that*

$$|\{i \in [n] \mid f(a_i) = b_i\}| \;\geq\; r \;>\; (d_x - 1) + (k-1) \cdot (d_y - 1)\,.$$

Then, $(y - P(x)) | R(x,y)$.

Proof: Let $I = \{i \in [n] \mid P(a_i) = y_i\}$. Then $R(a_i, P(a_i)) = 0$ for all $i \in I$. It follows that the univariate polynomial $R(x, P(x))$ has at least $|I|$ roots. But $R(x, R(x))$ has degree at most $(d_x - 1) + (k-1) \cdot (d_y - 1)$. Thus $Q(x, f(x)) \equiv 0$.

It follows that $y - P(x) | R(x,y)$. Indeed, we can write $R(x,y) = (y - P(x)) \cdot A(x,y) + B(x,y)$ where $\deg_y(B) < \deg_y(y - f(x)) = 1$. So $B(x,y)$ does not depend on y. Now $Q(x, f(x)) \equiv 0$ implies $B(x,y) = B(x) \equiv 0$. ■

It remains to choose the values of the parameters d_x, d_y and r to satisfy the conditions for the above lemmas. We can choose $d_x = \sqrt{n \cdot k}$ and $d_y = \sqrt{n/k} + 1$ and $r = 2\sqrt{n \cdot k}$ which satisfy the conditions above. Note that the list size equals $d_y - 1 = \sqrt{n/k}$. As an example, if $k = \varepsilon \cdot n$, we can tolerate an error rate of $1 - 2\sqrt{\varepsilon}$, while producing a list of size $\sqrt{1/\varepsilon}$.

2 Reed-Muller codes

One limitation of Reed-Solomon code is that it requires large field, in particular, $q \geq n$. Reed-Muller codes are generalization of Reed-Solomon codes that can be defined on any field size, even over $\mathbb{F}_2$.
Specifically, the Reed-Muller code $\mathrm{RM}_p(d, m)$ is a linear code over $\mathbb{F}_p$, where the message $(c_{i_1,\ldots,i_n})_{0 \leq i_1 + \cdots + i_n \leq d}$ is identified with the polynomial

$$P(\mathbf{x}) = P(x_1, \ldots, x_m) = \sum_{0 \leq i_1 + \cdots + i_m \leq r} c_{i_1,\ldots,i_m} x_1^{i_1} \cdots x_m^{i_m},$$

which is a multivariate polynomial of total degree at most d in m variables. $\mathrm{RM}_p(d, m)$ maps P to $\{P(\mathbf{x})\}_{\mathbf{x} \in \mathbb{F}_p^m}$, i.e. the codeword is the evaluation of P over all points in $\mathbb{F}_p^m$. We will discuss this code further in the next lecture.

References

[Gur07] Venkatesan Guruswami. Algorithmic results in list decoding. *Found. Trends Theor. Comput. Sci.*, 2(2):107–195, January 2007. URL: `http://dx.doi.org/10.1561/0400000007`, `doi:10.1561/0400000007`. 1

[Sud97] Madhu Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *J. Complexity*, 13(1):180–193, 1997. URL: `http://dx.doi.org/10.1006/jcom.1997.0439`, `doi:10.1006/jcom.1997.0439`. 1