CS225: Pseudorandomness Prof. Salil Vadhan

Lecture 15: List-Decoding Algorithms

April 5, 2007

Based on scribe notes by xxxx.

Let $\mathcal{C}$ be a code with encoding function $\mathrm{Enc} : \{1, \ldots, N\} \to \Sigma^{\hat{n}}$. Given any received word $r \in \Sigma^{\hat{n}}$, we would like to find all elements of $\mathrm{LIST}(r, \varepsilon) = \{m : \mathrm{agr}(m, r) \geq \varepsilon\}$ in polynomial time, where $\mathrm{agr}(m, r) = \Pr_y[m_y = r_y]$. (For convenience, we have switched to measuring the agreement ε instead of the list-decoding distance $\delta = 1 - \varepsilon$ as discussed last time.)

1 Review of Algebra

- For every prime power $q = p^k$ there is a field $\mathbb{F}_q$ of size q, and this field is unique up to isomorphism (renaming elements). The prime p is called the *characteristic* of the field. $\mathbb{F}_q$ has a description of length $O(\log q)$ enabling addition, multiplication, and division to be formed in polynomial time (i.e. time $\mathrm{poly}(\log q)$). If $q = p^k$ for a given prime p and integer k, this description can be found probabilistically in time $\mathrm{poly}(\log p, k) = \mathrm{poly}(\log q)$ and deterministically in time $\mathrm{poly}(p, k)$. Note that for even finding a prime p of a desired bitlength, we only know time $\mathrm{poly}(p)$ deterministic algorithms. Thus, for computational purposes, a convenient choice is often to instead take $p = 2$ and k large, in which case everything can be done deterministically in time $\mathrm{poly}(k) = \mathrm{poly}(\log q)$.
- For every field $\mathbb{F}$, $\mathbb{F}[X_1, \ldots, X_n]$ is the integral domain consisting of formal polynomials $Q(X_1, \ldots, X_n)$ with coefficients in $\mathbb{F}$, where addition and multiplication of polynomials is defined in the usual way.
- A polynomial $Q(X_1, \ldots, X_n)$ is *irreducible* if we cannot write $Q = RS$ where R, S are non-constant polynomials.
- $\mathbb{F}[X_1, \ldots, X_n]$ is a *unique factorization domain.* That is, every polynomial p can be factored as $Q = Q_1 Q_2 \cdots Q_m$, where each Q_i is irreducible and this factorization is unique up to reordering and multiplication by constants from $\mathbb{F}$. Given the description of a finite field $\mathbb{F}_{p^k}$ and the polynomial Q, this factorization can be done in probabilistically in time $\mathrm{poly}(\log p, k, |Q|)$ and deterministically in time $\mathrm{poly}(p, k, |Q|)$.
- For $Q(Y, Z) \in \mathbb{F}[Y, Z]$ and $f(Y) \in \mathbb{F}[Y]$, if $Q(Y, f(Y)) = 0$, then $Z - f(Y)$ is one of the irreducible factors of $Q(Y, Z)$ (and thus can be found in polynomial time).

2 List-Decoding Reed-Solomon Codes

Theorem 1 (Sudan) *There is a polynomial-time algorithm for decoding the Reed-Solomon code of degree d over $\mathbb{F}_q$ up to distance $\delta = 1 - 2\sqrt{d/q}$.*

In fact the constant of 2 can be improved to 1, matching the combinatorial list-decoding radius for Reed–Solomon codes given by an optimized form of the Johnson Bound, but we will not do this optimization here.

Proof: We are given a received word $r : \mathbb{F}_q \to \mathbb{F}_q$, and want to find all elements of $\mathrm{LIST}(r, \varepsilon)$ for $\varepsilon = 2\sqrt{d/q}$.

Step 1: Find a low-degree Q 'explaining' r. Specifically, $Q(Y, Z)$ will be a nonzero bivariate polynomial of degree at most d_Y in its first variable Y and d_Z in its second variable, and will satisfy $Q(y, r(y)) = 0$ for all $y \in \mathbb{F}_q$. Each such y imposes a linear constraint on the $(d_Y + 1)(d_Z + 1)$ coefficients of Q. Thus, this system has a nonzero solution provided $(d_Y + 1)(d_Z + 1) > q$, and it can be found in polynomial time by linear algebra (over $\mathbb{F}_q$).

Step 2: Argue that each $f(Y) \in \mathrm{LIST}(r)$ is a 'root' of Q. Specifically, it will be the case that $Q(Y, f(Y)) = 0$ for each $f \in \mathrm{LIST}(r, \varepsilon)$. The reason is that $Q(Y, f(Y))$ is a univariate polynomial of degree at most $d_Y + d \cdot d_Z$, and has at least εq zeroes (one for each place that f and r agree). Thus, we can conclude $Q(Y, f(Y)) = 0$ provided $\varepsilon q > d_Y + d \cdot d_Z$. Then we can enumerate all of the elements of $\mathrm{LIST}(r)$ by factoring $Q(Y, Z)$ and taking all the factors of the form $Z - f(Y)$.

For this algorithm to work, the two conditions we need to satisfy are

$$(d_Y + 1)(d_Z + 1) > q,$$

and

$$\varepsilon q > d_Y + d \cdot d_Z.$$

These conditions can be satisfied by setting $d_Y = \lfloor \varepsilon q/2 \rfloor$, $d_Z = \lfloor \varepsilon q/(2d) \rfloor$, and $\varepsilon = 2\sqrt{d/q}$. ■

Note that the rate of Reed-Solomon codes is $\rho = (d + 1)/q = \Theta(\varepsilon^2)$. The alphabet size is $q = \tilde{\Omega}(n/\rho) = \tilde{\Omega}(n/\varepsilon^2)$. In contrast, an optimal code would have $\rho \approx \varepsilon$ and $q = O(1/\varepsilon)$.

3 Parvaresh–Vardy Codes

Our aim is to improve the rate-distance tradeoff to $\rho = \tilde{\Theta}(\varepsilon)$. Intuitively, the power of the Reed–Solomon list-decoding algorithm comes from the fact that we can interpolate the q points $(y, r(y))$ of the received word using a *bivariate* polynomial Q to be of degree roughly $\sqrt{q}$ in each variable (think of $d = O(1)$ for now). If we could use m variables instead of 2, then the degrees would only have to be around $q^{1/m}$.

First attempt: Replace Step 1 with finding an $(m + 1)$-variate polynomial $Q(Y, Z_1, \ldots, Z_m)$ of degree d_Y in Y and d_Z in each Z_i such that $Q(y, r(y), r(y), \ldots, r(y)) = 0$ for every $y \in \mathbb{F}_q$.

Second attempt: Replace Step 1 with finding an $(m+1)$-variate polynomial $Q(Y, Z_1, \ldots, Z_m)$ of degree d_Y in Y and $d_Z = h - 1$ in each Z_i such that $Q(y, r(y)^h, r(y)^{h^2}, \ldots, r(y)^{h^{m-1}}) = 0$ for every $y \in \mathbb{F}_q$.

We get the best of both worlds by providing more information with each symbol — not just the evaluation of f at each point, but the evaluation of $m-1$ other polynomials, each of which is still of degree d (as is good for Step 1), but can be viewed as raising f to successive powers of h for the purposes of the getting a nonzero polynomial in one variable Z in Step 2.

To introduce this idea, we need some additional algebra.

- For univariate polynomials $f(Y)$ and $E(Y)$, we define $f(Y) \bmod E(Y)$ to be the remainder when f is divided by E. If $E(Y)$ is of degree k, then $f(Y) \bmod E(Y)$ is of degree at most $k-1$.
- The ring $\mathbb{F}[Y]/E(Y)$ consists of all polynomials of degree at most $k-1$ with arithmetic modulo $E(Y)$ (analogous to $\mathbb{Z}_n$ consisting integers smaller than n with arithmetic modulo n). If E is irreducible then, $\mathbb{F}[Y]/E(Y)$ is a field (analogous to $\mathbb{Z}_p$ being a field when p is prime). Indeed, this is how the finite field of size p^k is constructed: take $\mathbb{F} = \mathbb{Z}_p$ and $E(Y)$ to be an irreducible polynomial of degree k over $\mathbb{Z}_p$, and then $\mathbb{F}[Y]/E(Y)$ is the (unique) field of size p^k.
- A multivariate polynomial $Q(Y, Z_1, \ldots, Z_m)$ can be reduced modulo $E(Y)$ by writing it as a polynomial in variables $Z_1, \ldots, Z_m$ with coefficients in $\mathbb{F}[Y]$ and then reducing each coefficient modulo $E(Y)$.

Now we can define the Parvaresh–Vardy codes.

- $\Sigma = \mathbb{F}_q^m$ for the finite field $\mathbb{F}_q$ of size q and an integer parameter m.
- Blocklength: q.
- Message space: $\mathbb{F}_q^{d+1}$, where we view each message as representing a polynomial $f(Y)$ of degree at most d over $\mathbb{F}_q$.
- Codewords: for $y \in \mathbb{F}_q$, the y'th symbol of the encoding of f is

$$[f_0(y), f_1(y), \ldots, f_{m-1}(y)],$$

  where $f_i(Y) = f(Y)^{h^i} \bmod E(Y)$ and E is a fixed irreducible polynomial of degree $d+1$ over $\mathbb{F}_q$.

Theorem 2 *For an appropriate setting of h and m, the Parvaresh–Vardy code above has rate $\rho = \tilde{\Omega}(d/q)$ and can be list-decoded in polynomial time up to distance $\delta = 1 - \tilde{O}(d/q)$.*

Proof: We are given a received word $r : \mathbb{F}_q \to \mathbb{F}_q^m$.

Step 1: Find a low-degree Q 'explaining' r. We find a polynomial $Q(Y, Z_0, \ldots, Z_{m-1})$ of degree at most d_Y in its first variable Y and at most $h-1$ in each of the remaining variables, and will satisfy $Q(y, r(y)) = 0$ for all $y \in \mathbb{F}_q$.

This is possible provided

$$d_Y \cdot h^m > q.$$

Moreover, we may assume that Q is not divisible by $E(Y)$. If it is, we can divide out all the factors of $E(Y)$, which will not affect the conditions $Q(y, r(y)) = 0$ since E has no roots (being irreducible).

Step 2: Argue that each $f(Y) \in \mathrm{LIST}(r)$ is a 'root' of a related univariate polynomial Q^*. First, we argue as before that if $f \in \mathrm{LIST}(r, \varepsilon)$, we have

$$Q(Y, f_0(Y), \ldots, f_{m-1}(Y)) = 0.$$

This will be ensured provided

$$\varepsilon q > d_Y + (h-1) \cdot d \cdot m.$$

Once we have this, we can reduce both sides modulo $E(Y)$ and deduce

$$\begin{aligned} 0 &= Q(Y, f_0(Y), f_2(Y), \ldots, f_{m-1}(Y)) \bmod E(Y) \\ &= Q(Y, f(Y), f(Y)^2, \ldots, f(Y)^{m-1}) \bmod E(Y) \end{aligned}$$

Thus, if we define the univariate polynomial

$$Q^*(Z) = Q(Y, Z, Z^h, \ldots, Z^{h^{m-1}}) \bmod E(Y),$$

then $f(Y)$ is a root of Q^* over the field $\mathbb{F}_q[Y]/E(Y)$.

Observe that Q^* is nonzero because Q is not divisible by $E(Y)$ and has degree at most $h-1$ in each Z_i. Thus, we can find all elements of $\mathrm{LIST}(r)$ by factoring $Q^*(Z)$.

For this algorithm to work, the two conditions we need to satisfy are

$$d_Y \cdot h^m > q.$$

and

$$\varepsilon q > d_Y + (h-1) \cdot d \cdot m.$$

We can satisfy the second condition by setting $d_Y = \varepsilon q - dhm$, in which case the first condition is satisfied provided

$$\varepsilon > \frac{1}{h^m} + \frac{dhm}{q}.$$

The theorem can be obtained by taking $h = 2$ and $m = O(\log(1/\varepsilon))$, and noting that the rate is $\rho = d/(mq)$. ■

4 Folded Reed–Solomon Codes

We now sketch the ideas that were used by Guruswami and Rudra last year to achieve list-decoding capacity.

They use the Parvaresh–Vardy construction with $E(Y) = Y^{q-1} - \gamma$, where γ is generator of $\mathbb{F}_q^*$. (That is, $\{\gamma, \gamma^2, \ldots, \gamma^{q-1}\} = \mathbb{F}_q \setminus \{0\}$.) Then it turns out that $f^q(Y) = f(\gamma Y) \bmod E(Y)$. So they use $f_i(Y) = f^{q^i}(Y) \bmod E(Y)$, and for each nonzero element y of $\mathbb{F}_q$, the y'th symbol of the PV encoding of $f(Y)$ becomes

$$[f(y), f(\gamma y), \ldots, f(\gamma^{m-1} y)] = [f(\gamma^j, f(\gamma^{j+1}), \ldots, f(\gamma^{j+m-1})],$$

where we write $y = \gamma^j$.

Thus, the symbols of the encoding have a lot of overlap. For example, the γ^j'th symbol and the γ^{j+1}'th symbol share all but one component. Intuitively, this means that we should only have to send roughly a $1/m$ fraction of the symbols of the codeword, saving us a factor of m in the rate. (The other symbols can be automatically filled in by the receiver.) Thus, the rate becomes $\rho \approx d/q$, just like in Reed–Solomon codes.

However, there is still an extra factor m in the second term of

$$\varepsilon > \frac{1}{h^m} + \frac{dhm}{q}.$$

prohibit us to achieve $\rho = \Theta(\varepsilon)$. To deal with this, we don't just require that $Q(y, r(y)) = 0$ for each y, but instead require that Q has a root of multiplicity s at each point $(y, r(y))$. Formally, this means that the polynomial $Q(Y + y, Z_0 + r(y)_0, \ldots, Z_{m-1} + r(y)_{m-1})$ has no monomials of degree smaller than s.

Then the second inequality becomes

$$\varepsilon q s > d_Y + (h-1) \cdot d \cdot m.$$

However, we pay a price in the other condition, because asking for a root of multiplicity s amounts to $\binom{m+s}{s-1}$ constraints on the coefficients of Q (one for each monomial of degree smaller than s). So the other constraint becomes

$$d_Y \cdot h^m > q \cdot \binom{m+s}{s-1}.$$

If we take large $s = m$, these two constraints can be satisfied provided

$$\varepsilon > \frac{1}{m \cdot (h/4)^m} + \frac{dhm}{qs} \approx \frac{d}{q} \approx \rho,$$

as desired.