Lecture 7: List Decoding and Tree Codes

Anup Rao

October 16, 2019

Today, we explore two disjoint topics. The first is List-Decoding

List Decoding

In many applications it is enough to narrow down the transmitted codeword to a small list of candidates. This is called List-Decoding. The advantage of doing this is that we can hope to tolerate a much higher fraction of errors.

Indeed, one can show the following bound, that we do not prove here:

Theorem 1 (Johnson Bound)**.** *If $C \subseteq \Sigma^n$ is a code of distance d, then any Hamming ball of radius $n - \sqrt{n(n-d)}$ contains at most $n|\Sigma|$ codewords.*

In particular, this implies that the Reed-Solomon code of dimension k can be list decoded (with a list of size n^2) even if the number of errors is as large as $n - \sqrt{n(k+1)}$. Note that this number of errors far exceeds the distance of the code, which is $n - k + 1$.

Here we give an algorithm due to Sudan for this problem. Suppose we are working with a finite field $\mathbb{F}$ of size q, and we receive a string $\beta_1, \ldots, \beta_q$, that is supposed to be the evaluations of $f \in \mathbb{F}[X]$ of degree $k-1$ on the points $\alpha_1, \ldots, \alpha_q$, after some errors.

The idea of the algorithm is try and reconstruct the polynomial $Q(X, Y) = (Y - f_1(X))(Y - f_2(X)) \ldots$, where here f_1, f_2 are all the polynomials of degree $k-1$ that have high agreement with the given received word.

Consider the space of all polynomials $Q(X, Y)$ whose degree in X is at most $\lceil \sqrt{nk} \rceil$, and degree in Y is at most $\lceil \sqrt{n/k} \rceil$. The number of monomials in such a polynomial is more than n, so there must be at least one such polynomial that is non-zero and yet vanishes on all inputs $Q(\alpha_i, \beta_i) = 0$. So, we can use Gaussian elimination to find a non-zero polynomial Q as above.

Here we discuss how to list-decode from $n - 2\sqrt{nk}$ errors. In the exercises, we shall explore how to decode from $n - \sqrt{nk}$ errors by modifying this algorithm.

Lemma 2. *If $f(X)$ is a degree $k-1$ polynomial that agrees with the received word in at least $2\sqrt{nk} + k + 1$ locations, then $Y - f(X)$ must divide $Q(X, Y)$.*

Proof. Consider the polynomial $Q(X, f(X))$. This is a polynomial of degree at most $\lceil \sqrt{n/k} \rceil \cdot (k-1) + \lceil \sqrt{nk} \rceil \leq 2\sqrt{nk} + k$, yet it vanishes on $2\sqrt{nk} + k + 1$ inputs. So, it must be identically 0.

Now, view the polynomial $Q(X, Y)$ as univariate polynomial over Y, with coefficeints from the finite field $\mathbb{F}/(g(X))$, where g is an irreducible polynomial of degree k. Since $f(X)$ is a root of Q, $(Y - f(X))$ must divide $Q(X, Y)$. □

To complete the list decoding algorithm, factor Q. The number of factors of the form $Y - f(X)$ is at most $\sqrt{n/k}$, since the degree of Y is at most $\sqrt{n/k}$.

It is not trivial to factor polynomials, but there are efficient algorithms that can do it. We do not discuss them here.

Tree Codes

A TREE CODE is a combinatorial object that has even stronger properties than codes. They were invented by Schulman. Formally, a binary tree code is a labeling of the edges of the infinite binary tree with symbols from some alphabet Σ. Given two nodes u, v in the tree, let $l(u, v)$ denote the least common ancestor of u, v. Given two nodes a, b in the tree where a is an ancestor of b, let $c(a, b) \in \Sigma^*$ denote the string labeling the path from a to b. The tree code has relative distance δ if whenever you take two nodes u, v that are at the same depth of the tree, $\Delta(c(l(u, v), u), c(l(u, v), v)) \geq \delta \cdot |c(l(u, v), u)|$. One way to think about this is that the path from the root of the tree to u, and the path from the root to v can have a large common part. However, on all the parts of the paths that are not common, the symbols labeling the edges must have large distance.

We first observe that tree codes contain error-correcting codes in them. Indeed, to encode an n-bit message x, consider the labels in the tree that correspond to the string $x0^n$. Now, if $x \neq y$ are two distinct n-bit strings, then labels of $x0^n$ and $y0^n$ are guaranteed to disagree in δn coordinates by the definition of the tree code. So, the tree code defines an encoding into an error correcting code. Actually, the tree code provides something much stronger. The above encoding can be done *online*. To compute the first symbol of the encoding, you only need to know the first bit of the message, and so on. This means you can start transmitting the codeword even before you see the whole message.

Schulman used tree codes to correct errors in communication protocols. Intuitively, if Alice and Bob are having an interactive conversation, they should use tree codes to encode all of their messages. The advantage is that after a long time has passed Bob will be certain of what Alice said in the distant past. This is because if Bob decodes that Alice's messages correspond to u instead of v, then the only way this can happen is if the number of errors in the interval corresponding to the path $l(u, v), v$ exceeded $\delta/2$. So, $l(u, v)$ is probably

quite close to u, and the early part of the transmissions were likely decoded correctly.

Tree codes exist

HERE IS A SIMPLE random construction that gives a tree code. Let $\mathbb{F}$ be a finite field, and let $P \in \mathbb{F}[[X]]$ be a uniformly random power series over $\mathbb{F}$. This is simply an infinite list of coefficients $P_0, P_1, \ldots$. Every vertex/edge at depth n in the infinite binary tree corresponds to some binary string $u \in \{0,1\}^n$, which we interpret as a polynomial $u(X) \in \mathbb{F}[X]$ of degree $n-1$, whose coefficients are either 0 or 1. The first step in the tree corresponds to u_0, and the next corresponds to u_1. The label of the edge is defined to be the degree $n-1$ coefficient of $P(X) \cdot u(X)$.

Can we hope that a uniformly random labeling will give a tree code with high probability?

To see that this gives a tree code, consider two arbitrary n-bit strings u, v. Suppose u, v have a common prefix up to the mth bit. Then $u(X) - v(X)$ is divisible by X^{m-1}. So, it is enough to prove that for every polynomial $f(X)$ of degree $n-1$ with coefficients in $\{1, 0, -1\}$, and $f_0 = 1$, the first n coefficents of $f(X) \cdot P(X)$ have at least δn non-zeros. Fix $f(X)$. For a random power series $P(X)$, the n coefficients are all uniformly random and independent of each other. So, the expected number of 0's is n/q, and the probability of seeing $(1-\delta)n$ elements that are 0 is at most $2^{h(\delta)n} \cdot q^{-\delta n}$. There are only 3^n choices for the polynomial f, so by the union bound, the probability of not having this property is at most $2^{h(\delta)n} \cdot q^{-\delta n} \cdot 3^n$. Finally, we conclude that the probability we do not get a tree code of distance δ is at most $\sum_{n=1}^{\infty} 2^{h(\delta)n} \cdot q^{-\delta n} \cdot 3^n$, which is less than $1/100$ for q large enough.

We do not know of any explicitly encodable or decodable tree codes.