10/15/2003

Notes for Lecture 14

Today we see constructions of locally decodable codes with low query complexity. To construct such codes, we exploit the connection between locally decodable codes and private information retrieval.

14.1 Locally Decodable Codes vs Private Information Retrieval

Recall that a code $C : \Sigma^k \to \Sigma^n$ is (δ, p) *locally decodable* with q queries if there exists a decoding algorithm that, on input $i \in [k]$ and given oracle access to a word at distance at most δn from the code, recovers the ith bit of the message with probability p by making at most q oracle queries. Typically, we require that q be constant or polylogarithmic in k, δ and p constant, and attempt to obtain the best possible rate under these constraints. For this range of parameters, we would like to obtain codes with constant or polynomial rate. However, for constant q, the best known constructions fall far short of these goals, achieving only $n = 2^{O(qk^{1/(q-1)})}$.

One such construction is based on the connection between locally decodable codes and private information retrieval. A *private information retrieval scheme* is a system that shares a k bit "database" x among independent servers $S_1, \dots, S_q$. The user obtains an input $i \in [k]$ and wants to find out the ith bit of x, with a single round of communication, in such a manner that no server can tell the value i. More formally, we require that the distribution of the query sent to each server be independent of i. (A weaker requirement is that the distributions corresponding to various i be statistically close in the view of each server.)

Last time we sketched how a private information retrieval scheme with q servers (over alphabet Σ) yields a q query locally decodable code. There is also a connection in the other direction: Suppose we start with a q query locally decodable code C with a nonadaptive decoding algorithm. Moreover, we assume that the marginal distribution of each query is uniform. (For example, the local decoding algorithm for the Hadamard code has this property.) In the private private information retrieval scheme for database x, each server has access to the codeword $C(x)$. On input i, the user simulates the local decoding algorithm, using server S_j as an oracle for the jth query.

The requirement that the local decoding algorithm makes uniform queries is somewhat unnatural, but can be dropped at the expense of increasing the rate and reducing the success probability.

14.2 Constructing Locally Decodable Codes

Reed-Muller codes are, in some sense, almost locally decodable. Consider a Reed-Muller code over m variables of total degree d. Say we are interested in recovering the ath symbol of the codeword, given a word that is corrupted at δn locations, where $\delta < 1/2(d+1)$. Recall the following algorithm from last lecture:

1. Choose a random $b \in \mathbb{F}^m$ and let $l(t) = a + tb$.
2. Query the word at locations $l(1), l(2), \ldots, l(d+1)$.
3. Compute the unique univariate degree d polynomial p such that $p(t) = l(t)$ for $t = 1, 2, \ldots, d+1$. Return $p(0)$.

This algorithm makes $d+1$ queries, and each such query is uniformly distributed in $\mathbb{F}^m$. By a union bound, with probability at least $1/2$ none of the codeword symbols indexed by the queries are corrupted and the algorithm succeeds.

A code is *systematic* if the message appears as a substring of the codeword. If the Reed-Muller code were systematic, we could recover the ath symbol of the message by applying the above algorithm to the codeword entry corresponding to this symbol. Even though this is not the case, we show that a natural restriction of the Reed-Muller code yields a systematic code of comparable rate.

Suppose that $k \leq \binom{m}{d}$. We index each message entry by a unique d element subset S of $[m]$. Given a message $x = (x_S)_{S \subseteq [m]}$, we define the polynomial

$$p_x(z_1, \ldots, z_m) = \sum_{S:|S|=d} x_S \prod_{j \in S} z_j$$

over the field $\mathbb{F}$, where $|\mathbb{F}| < 2d$. The encoding $C(x)$ is obtained by evaluating p_x over all points in $\mathbb{F}^m$.

Since p_x has total degree d, the local decoding algorithm for the Reed-Muller code also works for the code C. Moreover, C is systematic: We can write $x_S = p_x(e_S)$, where $e_S = (e_{S1}, \ldots, e_{Sm}) \in \mathbb{F}^m$ and

$$e_{Sj} = \begin{cases} 1 & \text{if } j \in S, \\ 0 & \text{otherwise.} \end{cases}$$

The code C can encode messages up to length $\binom{m}{d}$ and has codeword length $n = |\mathbb{F}|^m$, provided $|\mathbb{F}| > d$. If we take, say, $|\mathbb{F}| \leq 2d$, this yields $n = 2^{O(d \log dk^{1/d})}$, where $d = q - 1$.

14.3 Locally Decodable Codes over the Binary Alphabet

We modify the construction of C from the previous section to obtain locally decodable codes over binary alphabet. Our first step will be the construction of a q server private information retrieval scheme. Let $\mathbb{F} = \mathbb{F}_2$, $q = d+1$, and p_x and C be as above. Each server has access to the codeword $C(x)$.

On input S, the user chooses $z^1, \ldots, z^{d+1}$ independently at random from $\mathbb{F}^m$, conditional on $z^1 + \ldots + z^{d+1} = e_S$. Note that these vectors are d-wise independent. The user sends server S_j the vectors $z^{j'}$, for all $j' \neq j$. In the server's view, this is just a collection of dm random bits.

The user is interested in recovering the value $p_x(e_S) = p_x(z^1 + \ldots + z^{d+1})$. We view this as a formal polynomial over $\mathbb{F}$ with variables z_i^j, $i \in [m], j \in [d+1]$. Say a *monomial* M in $\mathbb{F}[z_i^j]$ is *computable* by server S_j if none of the variables $z_1^j, \ldots, z_m^j$ appears in M.

Let us consider the canonical expansion of $p_x(z^1 + \ldots + z^{d+1})$ over $\mathbb{F}[z_i^j]$. Note that the polynomial has degree d even over the new variables. Let M be a monomial of this expansion, also of degree d. By a pigeonhole argument, there must exist a j such that none of the variables $z_1^j, \ldots, z_m^j$ appears in M, namely, M is computable by S_j.

This suggests the following protocol for server S_j:

1. Compute the expansion of $p_x(z^1 + \ldots + z^{d+1})$ as a polynomial in $\mathbb{F}[z_i^j]_{i \in [m], j \in [d+1]}$.

2. Let A_j be the formal sum of monomials in the expansion that are computable by S_j but not computable by any of $S_1, \ldots, S_{j-1}$. Calculate A, substituting the bits sent by the user for the corresponding formal variables in A_j. Send the resulting value A_j^* to the user.

By construction, each monomial M appears in exactly one of the formal sums A_j, so that $p_x(e_S) = A_1^* + \ldots + A_{d+1}^*$. The user recovers x_S by adding up the values received from all the servers.

We are now one step away from a binary locally decodable code: The codeword will be a concatenation of $d+1$ binary strings $a_1, \ldots, a_{d+1}$, where a_j is a table of answers to all 2^{md} possible queries sent to server S_j. This gives codeword length $n = (d+1)2^{md} = q2^{O(qk^{1/(q-1)})}$.

On input i, the decoding algorithm computes the set of messages $M_1, \ldots, M_{d+1}$ for the private information retrieval scheme, permutes these in a random order and looks up the corresponding entries in the possibly corrupted codeword. The decoder outputs the sum of the looked up entries.

Since each M_j is a random string of length md, it is uniformly distributed over a_j, so after the random permutation each query is uniformly distributed over all possible entries. If the codeword is corrupted at fewer than δn places, with probability at least $1 - \delta q$ none of these are accessed by the decoding algorithm and the decoding succeeds.