6.897 Algorithmic Introduction to Coding Theory November 6, 2002

Lecture 17

Lecturer: Madhu Sudan *Scribe: Luis Rademacher*

Overview:

- Linear time encodable and decodable codes.
- Shannon capacity with linear time algorithms.

1 Introduction

Graph-based codes were presented in the previous lecture. A linear time decoding algorithm was presented for this codes, but encoding may take quadratic time. Now, we are interested in efficient, linear time encodable and decodable codes. Analogous to the idea of sparse parity-check matrices, the encoding could be faster if the generator matrix were sparse, but in this case it can't be a good error correcting code: if the generator matrix has at most c ones per row (for small c), then changing i^{th} bit of the message changes at most c bits of its encoding.

Spielman's family of codes achieves linear time encoding and decoding, and will be described in the following notes.

2 Error reducing code

A sparse generator matrix cannot provide a good error correcting code but can provide an error reducing code, that is, and encoding and decoding procedure such that, under suitable hypothesis, can reduce the number of errors (but not necessarily correct all of them). In this case, it is possible to get linear time encoding and decoding.

For a message length k, instead of specifying directly the generator matrix, we will specify a bipartite graph G with k left nodes and $k/2$ right nodes. The associated generator matrix will be:

$$G := \begin{bmatrix} I & A_g \end{bmatrix},$$

where I is the $k \times k$ identity and A_g is the $k \times k/2$ adjacency matrix of the graph G. The first k bits of the encoding are the message itself, the last $k/2$ bits are the check bits. This code will be denoted R_k.

Note that there is a fundamental difference between the meaning of this graphs and the graphs in the previous lecture. In both cases the left nodes are labelled with the bits of the message. Previously, the right nodes represented constraints over the left nodes, in this case the right nodes are labelled with check bits that are computed from the parity of their (right) neighbors.

If the degree of the left nodes is at most c, then the minimum distance is at most $c + 1$ (a one-bit change in the message flips 1 message bit and at most c check bits).

It is easy to see that this codes can be encoded in time $O(k)$. If we use a slightly modified version of the FLIP algorithm of the previous lecture, we can achieve linear time error-reduction. In terms of the previously defined FLIP algorithm, an unsatisfied right vertex in this case is one with a label different to the parity of its neighbors.

The following lemma states in a precise way the error-reducing capability of R_k:

Lemma 1 *Assume that the bipartite graph G, with k right nodes, is (c, c')-regular and (γ, δ)-expander, and suppose that the original encoding, (m, c), and the encoding with errors, (x, y), are (a, b)-close, that is $\Delta(m, x) \leq a$ and $\Delta(c, y) \leq b$. Suppose also that $a(c + 1) + b \leq \delta k$, $\gamma \geq 7c/8$, and $c \geq 8$. Then* FLIP *terminates and outputs a word that differs from m in at most $b/2$ bits. Moreover,* FLIP *takes time $O(k)$.*

Proof Denote $S = \{i : m_i \neq x_i\}$, $T = \{i : c_i \neq y_i\}$. For every iteration, the algorithm FLIP decreases in at least one the number of unsatisfied constraints, which is at most $ca+b$. Then, the algorithm finishes with some word x' such that $\Delta(x', m) \leq \Delta(x, m) + ca + b \leq a + ca + b \leq \delta k$. Denote $S' = \{i : m_i \neq x'_i\}$, then $|S'| \leq \delta k$. Because G is an expander, $|\Gamma_1(S')| \geq (2\gamma - c)|S|$. Denote U the set of unsatisfied right nodes, at the end of the algorithm. In this situation, every left vertex has at most $c/2$ unsatisfied right neighbors, thus $|U| \leq |S'|c/2$. On the other hand, $\Gamma_1(S') - T \subseteq U$, then $|U| \leq (2\gamma - c)|S'| - b$. The combination of both inequalities for $|U|$ implies $(2\gamma - 3c/2)|S'| \leq b$. This inequality and the hypotheses give immediately $|S'| \leq b/2$. ■

3 Spielman's error-correcting code

Denote C_k the Spielman's code for messages of length k. This code will be defined recursively in terms of error-reducing codes and Spielman's codes for shorter messages. The intuitive idea is to take into account the fact that the codes R_k can be error-correcting and efficient if we are able to protect the check bits from errors. A key step in achieving this is to encode the check bits with a Spielman's code of smaller message length.

More precisely, given a message of length k, its C_k-encoding has length $4k$. The first $3k/2$ bits of the encoding are the result of R_k applied to the message, that is, the first k bits of the encoding are the message itself (this bits are denoted M_k) and the next $k/2$ bits are the check bits (denoted A_k). This $k/2$ check bits are encoded with $C_{k/2}$ to provide the next $3k/2$ check bits (denoted B_k). Finally, (A_k, B_k), the $2k$ bits that are the result of this $C_{k/2}$-encoding, are encoded with R_{2k} to provide k additional check bits (denoted D_k). The rate of C_k is $1/4$.

For the encoding time of C_k, it is clear that R_k-encoding takes time $O(k)$. Then, if $T_E(k)$ is the encoding time of C_k, we have

$$T_E(k) \leq O(k) + T_E(\frac{k}{2}) + O(k).$$

This implies that $T_E(k) = O(k)$.

A decoding algorithm for C_k is:

1. Run FLIP on (A_k, B_k, D_k) (the last $3k$ bits) until it stops to obtain (A'_k, B'_k).
2. Recursively decode the result of the previous step (A'_k, B'_k) with C_{k-1} to obtain A''_k.
3. Run FLIP on A''_k, the result of the previous step, to recover message.

For the decoding time of C_k, steps 1 and 3 take linear time. Then, if $T_D(k)$ is the decoding time of C_k, we have:

$$T_D(k) \leq O(k) + T_E(\frac{k}{2}) + O(k).$$

This implies that $T_E(k) = O(k)$.

About the effectiveness of the decoding of C_k, suppose that we allow at most ϵk errors. After step 1, the number of errors in (A'_k, B'_k) is less that or equal to $\frac{\epsilon k}{2}$. By inductive hypothesis, this implies that step 2 will recover B_k without errors. Because this are the check bits of R_k for the original message, this implies that algorithm FLIP will recover the message M_k without errors in step 3.[1]

4 Modification to Spielman's codes to handle a tiny fraction of errors with small overhead

It is possible to modify Spielman's codes to have a rate better than $1/4$, if we reduce the error-correction capability appropriately, while we retain the linear encoding and decoding property. The idea is to

[1] In fact, the same argument and conclusion applies if we allow $3\epsilon k$ errors, distributed in the following way: ϵk errors in M_k, the message, ϵk errors in (A_k, B_k), and ϵk errors in C_k.

compute few check bits in the first step: instead of $k/2$ check bits we consider αk bits, for small α, $0 < \alpha < 1$. Then we protect this check bits with $C_{\alpha k}$, and add additional check bits as in the low-rate case. This code has an $\epsilon > 0$ fraction of errors correcting capability, such that $\alpha \to 0$ as $\epsilon \to 0$.

5 The Random Errors Case

Consider now the case of random errors that flip every bit independently with probability p, that is, BSC_p. We will construct a code based on Forney's idea, that is, composition, but instead of considering the Reed-Solomon codes as the outer code we will consider a high-rate Spielman's code (as described in the previous section), that can correct an ϵ fraction of (adversarial) errors and has rate $1 + f(\epsilon)$, with $f(\epsilon) \to 0$ as $\epsilon \to 0$. The inner code takes blocks of length c (independent of the message length k) and gives encoded blocks of length c', $c \approx (1 - H(p))c'$. The encoded length is

$$n := \frac{k(1 + f(\epsilon))}{1 - H(p)}.$$

The inner code is decoded by maximum likelihood decoding, that is, in constant time. Because of a Chernoff-bound argument (or Shannon's theorem), the probability of decoding error in a particular block after inner decoding is very small. Thus, with high probability the outer code is able to correct this errors.

The main conclusion of this argument is that we can achieve correction of random errors up to Shannon's bound with linear time encoding and decoding.

References

[1] Daniel A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Trans. Inform. Theory*, 42(6, part 1):1723–1731, 1996. Codes and complexity.