Lecture 16

Tornado Codes

16.1. Introduction

Tornado codes are a type of Forward Error Correcting (FEC) code. They were designed to correct erasures with high probability. The encoding and decoding algorithms for these codes have fast and simple software implementations.

The design of Tornado codes requires the construction of a random irregular bipartite graph with a carefully chosen degree sequence. The encoding and decoding algorithm are very similar. Each requires an XOR operation per edge of the graph. The graph is chosen to be sparse so that encoding and decoding algorithm are fast. The decoding algorithm can be modeled as set of differential equations and the solution to these equations can then be expressed as a polynomial in one variable whose coefficients determine the degree sequence of the graph. The operation of one such graph is shown in Fig.16.1. The nodes on the left are known. The values of nodes on the right are computed by performing an XOR operation of the neighboring input nodes. The graph $B \in G(\lambda, \rho, n)$ has n message bits as input and produces βn check bits. The codeword consists of the n message bits and all the check bits produced at each stage of the cascade. It is thus a systematic code.

16.2. Using differential equation to model the decoding process

The process starts with the initial random graph B, with n nodes on left and βn nodes on right. Consider the two vectors (λ_i) and (ρ_i) that are the fractions of edges of degree i on the left and right respectively, with respect to the total number E of edges in the orginal graph. Let a_l be the average node degree on the left and a_r be the average node degree on the right. Initially we can write,

$$n = \sum_i \frac{\lambda_i E}{i}$$

and

$$a_l = \frac{E}{n}$$

therefore

$$a_l^{-1} = \frac{n}{E} = \sum_i \frac{\lambda_i}{i} = \int_0^1 \lambda(x) dx$$

Lecture by Amin Shokrollahi
Scribes by Mahdi Jafari

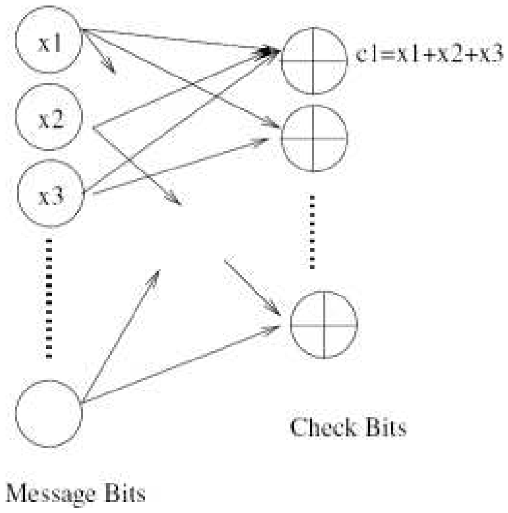

Figure 16.1: Irregular Bipartite Graph.

Similarly on the right we have,

$$\beta m = \sum_i \frac{\rho_i E}{i}$$

and

$$a_r = \frac{E}{\beta n}$$

so

$$a_r^{-1} = \frac{\beta n}{E} = \sum_i \frac{\rho_i}{i} = \int_0^1 \rho(x)dx$$

Despite the fact that the decoding process is discrete, it is going to be modeled using differential equations in the variable t. The passage of time, variable t, scaled in such a way that each time unit of length $\Delta t = \frac{1}{E}$ corresponds to one step of the decoding process. Let δ be the fraction of losses in the message. At the start of the process, just prior to time 0, each node on the left is removed with probability $(1-\delta)$, because this symbol is received without error and is therefore removed from the subgraph. So the initial subgraph of B contains δn nodes on the left. If the decoding process completes successfully, it runs until time $\delta n \Delta t = \frac{\delta}{a_l}$. Let define the following variables,
$l_i(t) =$ fraction of edges of degree i on the left at time t (with repect to E),
$r_i(t) =$ fraction of edges of degree i on the right at time t (with repect to E).
Let,

$$e(t) = \sum_i l_i(t) = \sum_i r_i(t)$$

Which is the fraction of edges (in term of E) remaining at time t. At each step, a random node of degree one on the right is selected and the corresponding neighbour node on the left and all of its adjacent edges are deleted. The process necessarily terminated if there is no such a node on the right. The probability that the edge adjacent to the node of degree one on the right has degree i on the left is $\frac{l_i(t)}{e(t)}$. After finding the value for the left node and removing it we lose i edges of degree i on the left. So the difference equation for the expected change in the number of edges, $L_i(t)$, of degree i on the left can be expressed as,

$$L_i(t+\Delta t) - L_i(t) = -\frac{il_i(t)}{e(t)}$$

and it should be known that $l_i(t) = \frac{L_i(t)}{E} = L_i(t)\Delta t$. Taking the limit $E \to \infty$, the difference equation become the following differential equation,

$$\frac{dl_i(t)}{dt} = -\frac{il_i(t)}{e(t)}$$

The expected number of removed edges in each step, after removing a left node, is $a(t) = \sum \frac{il_i(t)}{e(t)}$. When we remove a node of degree i on the left, we remove the one edge of degree one from the right along with the $i-1$ edges adjacent to this node. So on the average the number of other edges (other than the edge with degree one) deleted is $a(t)-1$. The right endpoitns of these $i-1$ other edges are randomly distributed. If we remove one of the edge of degree j on the right, we lose j edges of degree j and gain $j-1$ edges of degree $j-1$. The probability that an edge has degree j on the right is $\frac{r_j(t)}{e(t)}$. For $i > 1$, the difference equation

$$R_i(t+\Delta t) - R_i(t) = (r_{i+1}(t) - r_i(t))\frac{i(a(t)-1)}{e(t)}$$

describes the expected change in the number of edges, $R_i(t)$, of degree i on the right. Taking the limit $E \to \infty$, the corresponding differential equations for the $r_i(t)$ would be

$$\frac{r_i(t)}{dt} = (r_{i+1}(t) - r_i(t))\frac{i(a(t)-1)}{e(t)}$$

The differential equation for $i = 1$ is different from above because at each step an edge of degree 1 is removed from the right. So the differential equation for $r(t)_1$ is

$$\frac{r_1(t)}{dt} = (r_2(t) - r_1(t))\frac{i(a(t)-1)}{e(t)} - 1$$

We are interested in the values of $r_1(t)$ as a function of t. As long as $r_1(t) > 0$ there is a node of degree one on the right and the decoding process continues. When $r_1(t) = 0$ the process stops. So our aim is $r_1(t) > 0$ until all the nodes on the left are deleted and the process finished successfully. The solution to these set of differential equations can be expressed using the degree sequence functions

$$\lambda(x) = \sum_{i \geq 1} \lambda_i x^{i-1}$$

and

$$\rho(x) = \sum_{i \geq 1} \rho_i x^{i-1}$$

Lemma 16.1. *The solution $r_1(t)$ of the above differential equation is given by*

$$r_1(x) = \delta\lambda(x)[x - 1 + \rho(1 - \delta\lambda(x))] \tag{16.1}$$

From 16.1 and the fact that we need $r_1(x) > 0$ untill all the left nodes are deleted we can conclude that the following condition is required

$$\rho(1 - \delta\lambda(x)) > 1 - x, \quad x \in (0, 1] \tag{16.2}$$

16.3. From differential equations to build codes

As stated in the previous parts the differential equations describe the limiting behavior of the decoding process on the graph B, as the block length goes to infinity. From this one can derive that if 16.2 is violated so the process fails for large block lengths with high probability. Proving the process progresses almost to completion if 16.2 is satisfied is possible by relating the differential equations to the underlying random process. Proving the process runs to completion can handle with a separate combinatorial argument as follow

Lemma 16.2. *Let B be a bipartite graph with n message bits that is chosen at random with edge degrees specified by $\lambda(x)$ and $\rho(x)$, $B \in G(\lambda, \rho, n)$. Let δ be fixed so that*

$$\rho(1 - \delta\lambda(x)) > 1 - x, \quad x \in (0, 1]$$

For all $\eta > 0$ there is some n_0 such that for all $n \geq n_0$, if the message bits of $\mathcal{C}(B)$ are lost independently with probability δ, then with probability at least $1 - n^{2/3} exp(-\sqrt[3]{n}/2)$ the recovery algorithm terminates with at most ηn message bits erased.

The following combinatorial lemma is useful in showing that the process termiantes successfully when there are no nodes on the left of degree one or two.

Lemma 16.3. *Let B be a bipartite graph with n left nodes chosen at random with edge degrees specified by $\lambda(x)$ and $\rho(x)$, such that $\lambda(x)$ has $\lambda_1 = \lambda_2 = 0$. Then there is some $\eta > 0$, such that with probability 1-O(1) if at most an η fraction of the nodes on the left in B remain, then the process terminates successfully. In another words*

$$\exists \eta > 0 : Pr[\forall S \subseteq L, |s| \leq \eta n : |E(S)| < 2\Gamma(S)] \geq const$$

Proof. Let S be any set of nodes on the left of size at most ηn, where will be chosen later. Let a be the average degree of these nodes, $a = \frac{|E(S)|}{|S|}$. If the number of nodes on the right that are neighbors of S is greater than $\frac{a|S|}{2}$, then one of these nodes has only one neighbor in S, and so the process can continue. Thus, we only need to show that the initial graph is a good expander on small sets. Let Ev_s denote the event that a subset of size s of the nodes on the left has at most $as/2$ neighbors. We first bound $Pr(Ev_s)$, and then sum $Pr(Ev_s)$ over all values of s no larger than ηn. Fix any subset S of the left nodes of size s, and any subset T of the right nodes of size $as/2$. There are $\binom{n}{s}$ ways of choosing S, and $\binom{\beta n}{as/2}$ ways of choosing T. The probability that T contains all as the neighbors of the vertices in S is $(\frac{|T|}{\beta n})^{as}$, where $|T| = as/2$. Hence, we have

$$Pr(Ev_s) \leq \binom{n}{s}\binom{\beta n}{as/2}\left(\frac{as}{2\beta n}\right)^{as}$$

Note that $\binom{n}{k} \leq (ne/k)^k$, So we have

$$Pr(Ev_s) \leq \left(\frac{s}{n}\right)^{(a/2-1)s} c^s \leq \left(\frac{sc^2}{n}\right)^{s/2}$$

where c is a constant (depending on β and a). Since the graph does not have nodes of degree one or two, we ahve that $Pr(Ev_1) = Pr(Ev_2) = 0$. Choosing $\eta \leq 1/(2c^2)$ yields

$$\begin{aligned}\sum_{s=1}^{\eta n} Pr(Ev_s) &\leq \sum_{s=3}^{\eta n}\left(\frac{sc^2}{n}\right)^{s/2} \leq \frac{3c^2}{n\sqrt{n}} + \sum_{s=4}^{\eta n}\frac{1}{2^s} \\ &= O\left(\frac{1}{n\sqrt{n}}\right)\end{aligned}$$

which shows that, with high probability, the original graph is an expander on small subsets.

□

16.4. Capacity Achieving codes

In the sequel we will illustrate the construction of families of codes with linear time erasure decoding algorithms for any erasure probability p that can correct any p-fraction of erasures and whose rates come arbitrarily close to the capacity $1 - p$ of the erasure channel. In other words, we will show the construction of codes that are close to optimal in terms of their erasure recovery rate, and have linear time encoding and decoding algorithms. This will be done by finding an infinite family of solutions to the differential equations of Sec.16.2. in which δ is close to $1 - r$, where r is the rate. The Eq.16.2 is very handy if one wants to analyse the performance of random graphs with a given distribution. But, this condition does not give a clue on how to design good degree distributions λ and ρ. The aim is to construct sequences that asymptotically achieve the capacity of the erasure channel.

To make this definition more rigorous, we call a sequence (λ_n, ρ_n) capacity-achieving of rate r if for all $\delta < 1 - r$ there exits $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$ we have

$$\rho_n(1 - \delta\lambda_n(1 - x)) > x, \forall x \in (0, 1)$$

Note that any δ satisfying the inequality above can be at most $1 - r$. This can be followed by integration as follows. From Eq.16.2 we can write

$$\begin{aligned} \rho(1 - \delta\lambda(1 - x)) &> x \\ 1 - \delta\lambda(1 - x) &> \rho^{-1}(x) \\ \delta\lambda(x) &< 1 - \rho^{-1}(1 - x) \\ \int_0^1 \delta\lambda(x)dx &\leq 1 - \int_0^1 \rho^{-1}(x)dx \end{aligned}$$

suppose that $\rho(0) = 0$ so we have

$$\int_0^1 \rho^{-1}(x)dx = 1 - \int_0^1 \rho(x)dx$$

Therefore we get

$$\delta \leq \frac{a_l}{a_r} = 1 - r$$

To describe the sequence we do as follow. Let B be a bipartite graph with n left nodes and βn right nodes. The code will be described by the choice for the left and right degree sequences of B that satisfy condition 16.2. Let D be a positive integer that is used to trade off the average degree with how well the decoding process works, i.e., how close we can make δ to $\beta = 1 - r$ and still have the process finish successfully most of the time.

The left degree sequence is described by the following truncated heavy tail distribution. Let $H(D) = \sum_{i=1}^{D} 1/i$ be the harmonic sum truncated at D, and thus $H(D) \sim ln(D)$. Then, for all $i = 2, \ldots, D + 1$, the fraction of edges of degree i on the left is given by

$$\lambda_i := \frac{1}{H(D)(i - 1)}$$

So the polynomial $\lambda(x)$ become

$$\lambda(x) = \frac{1}{H(D)} \sum_{i=1}^{D} \frac{x^i}{i}$$

The average left degree a_l equals to $H(D)(D + 1)/D$. Recall that it is required that the average right degree a_r to satisfy $a_r = a_l/\beta$. The right degree sequence is defined by the Poisson distribution with mean a_r: for all $i \geq 1$ the fraction of edges of degree i on the right equals

$$\rho_i := \frac{e^{-\alpha}\alpha^{i-1}}{(i - 1)!}$$

where α is chosen to guarantee that the average degree on the right is a_r. In other words, α satisfies $\alpha e^{\alpha}/e^{\alpha} - 1 = a_r$.

Note that we allow $\rho_i > 0$ for all $i \geq 1$, and hence $\rho(x)$ is not truly a polynomial, but a power series. However, truncating the power series $\rho(x)$ at a sufficiently high term gives a finite distribution of the edge degrees for which the next lemma is still valid.

We show that when $\delta = \beta(1 - 1/D)$, then condition 16.2 is satisfied, i.e., $\rho(1 - \delta\lambda(x))) > 1 - x$ on $(0, 1]$. Note that $\lambda(x)$ is the expansion of $-ln(1 - x)$ truncated at the Dth term, and scaled so that $\lambda(1) = 1$. Further, $\rho(x) = e^{\alpha(x-1)}$.

Lemma 16.4. *With the above choices for $\rho(x)$ and $\lambda(x)$ we have $\rho(1-\delta\lambda(x)) > 1-x$ on (0,1] if $\delta \leq \beta/(1+1/D)$.*

Proof. Since $\rho(x)$ increases monotonically in x, we have

$$\rho(1-\delta\lambda(x))) > \rho(1+\delta ln(1-x)/H(D)) = (1-x)^{\alpha\delta/H(D)}$$

As $a_l = H(D)(1+1/D)$ and $a_r = a_l/\beta$, we obtain

$$\alpha\delta/H(D) = (1-e^{-\alpha})(1+1/D)\delta/\beta < \delta(1+1/D)/\beta \leq 1$$

which shows that the right-hand side of the above inequality is larger than $1-x$ on (0,1].

□

A problem is that Lemma 16.3 does not apply to this system because there are nodes of degree two on the left. To see this we note that the left degree distribution from the node perspective is

$$L(x) = \frac{\int_0^x \lambda(z)dz}{\int_0^1 \lambda(z)dz}$$

Assuming D is very larg we have

$$L(x) = \frac{1}{2}x^2 + \frac{1}{6}x^3 + \cdots + \frac{1}{i(i-1)}x^i + \ldots$$

To overcome this problem, we make a small change in the structure of the graph B. Let $\gamma = \beta/D^2$. We split the βn right nodes of B into two distinct sets, the first set consisting of $(\beta-\gamma)n$ nodes and the second set consisting of γn nodes. The graph B is then formed by taking the union of two graphs, B_1 and B_2. B_1 is formed as described up to this point between the n left nodes and the first set of $(\beta-\gamma)n$ right nodes. B_2 is formed between the n left nodes and the second set of γn right nodes, where each of the n left nodes has degree three and the $3n$ edges are connected randomly to the γn right nodes.

Lemma 16.5. *Let B be the bipartite graph described above. Then, with probability $1-O(n^{-3/2})$, the decoding process terminates successfully when started on a subgraph of B induced by δn of the left nodes and all βn of the right nodes, where $\delta = \beta(1-1/D)$.*

Proof. In the analysis of the process, we may think of B_2 as being held in reserve to handle nodes not already dealt with using B_1. First, using the same method as in Lemma 16.3 we can prove that there is some η such that a set S of $s \leq \eta n$ left nodes in the graph B_2 expands to a set of at least $3s/2$ nodes on the right, with probability $1-O(1/n^{3/2})$. (Note that all nodes on the left have degree three in this graph.) Combining Lemma 16.2 and Lemma 16.4, we see that the recovery process started on B_1 terminates with less than ηn nodes on the left unrecovered, with probability $1-O(exp(-n^a))$ for some positive a: note that the ratio of the number of left nodes to the number of right nodes in the graph B_2 equals $\beta(1-1/D^2)$, hence the condition in Lemma 16.4 translates to

$$\delta \leq \beta(1-1/D^2)/(1+1/D) = \beta(1-1/D)$$

which is obviously true. By the aforementioned expansion property of the subgraph of B_2 induced by the set of unrecovered left nodes, we see that the process terminates successfully.

□

Note that the degree of each left node in this modified construction of B is at most three bigger than the average degree of each left node in the construction of B described at the beginning of this section.