CS 229r Essential Coding Theory, Lecture 14	Mar 9, 2017

Lecture 14

Instructor: Madhu Sudan *Scribe: Zhixian Lei*

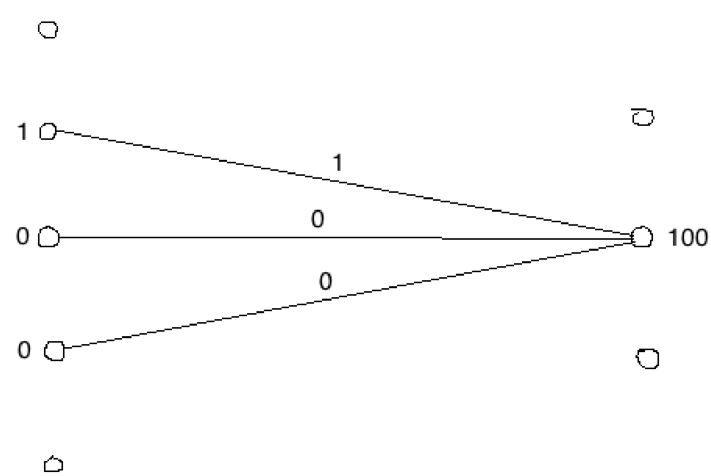
1 Overview

In this lecture we will review the ABNNR code by Alon, Brooks, Naor, Naor, and Roth (1992) and a generalization AEL code by Alon, Edmonds, and Luby (1995). The Guruswami-Indyk algorithm (2004) for decoding is also presented.

2 ABNNR code

We want to construct a graphically generated codes with good distance versus rate $R \sim 1 - \delta$ over large constant sized alphabet. ABNNR code gives $R = \Omega(1 - \delta)$. To see the construction of ABNNR code, we first a simple code by bipartite regular expander of n vertexes on both side and degree d for both side.

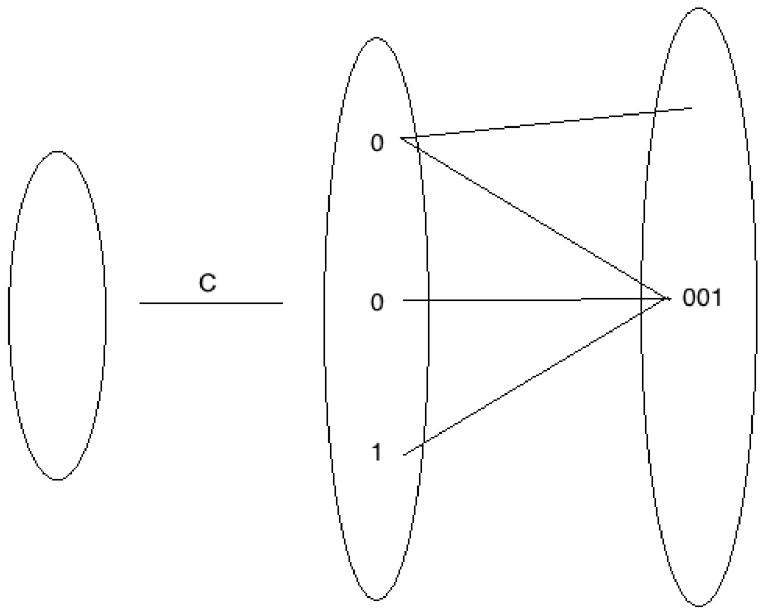
2.1 Simple code



This is a simple code. The message is assigned on left vertexes, and the encoding algorithm is just move bits to right vertexes and concatenate all bits on edges to get symbol on right.

Now we analyze the rate and distance of this simple code. Suppose the message is Σ^n, and the encoding is $(\Sigma^d)^n$ where Σ^d is the alphabet for the code. Then the rate is $1/d$. The distance depends on the expansion of the bipartite graph, but is at most d because if we change 1 bit of the message, only d places will change. Next we show ABNNR code

2.2 ABNNR code



The construction of ABNNR code has two steps. b) Use the simple code to get final codeword. The formal construction of ABNNR code is

1. Use some decent code C, for example, Justesen code to encode message in Σ^k to word in Σ^n
2. Assign each of the n bits to the left vertices of expander graph
3. Each of right vertex has d neighbors, assign each right vertex a d-bit string derived from its d neighbors. The n elements on the right will be our desired codeword.

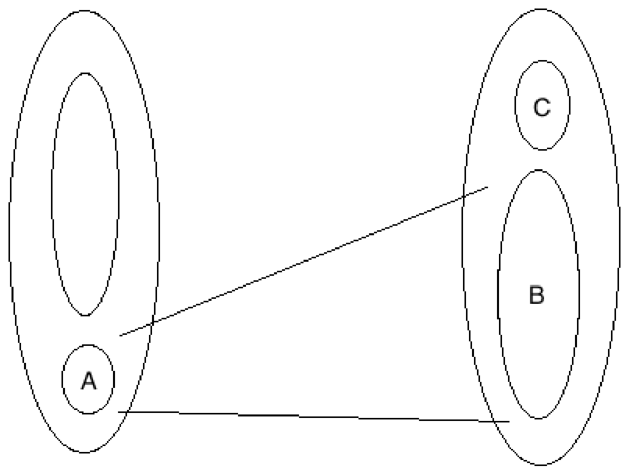
The the rate of ABNNR code is $\frac{k}{n}\frac{1}{d} = \frac{k}{nd}$. The distance of the code is decided by the bipartite graph. We have already assume that the bipartite graph has both left and right degree d.

Definition 1 ((α, β) expander)**.** *A bipartite graph is an* (α, β) *expander if for all set of size less than* ϵn *on the left expand by a factor* αd

If we use (α, ϵ) in the construction, the δ fraction of change in the middle will change $\alpha d\delta$ in the codeword. If the encoding C has relative distance δ, the relative distance of the entire is $\alpha d\delta$. We can also give an upper bound for this relative distance.

Suppose $\alpha \to 1$. A is a δ fraction of left set, and A expands to B. C is the rest of the right set excluding B. Then A and C are disjoint. C expands at most $(1-\delta)$ fraction of A, so $|C| \sim \frac{1}{d}$ and $|B| \sim 1 - 1/d$. So the distance of the code is $\sim 1 - \frac{1}{d}$ so we get

Theorem 2. *ABNNR achieves distance* $1 - \frac{1}{d}$ *with rate* $\Omega(\frac{1}{d})$*, which achieves near Singleton bound over large but constant alphabet*



The ABNNR code is pretty cool but it has a low rate. Next we will show AEL code which achieves constant rate.

3 AEL code

AEL code achieves $R = 1 - \delta - \epsilon$ for all $\epsilon > 0$ First we give the construction.

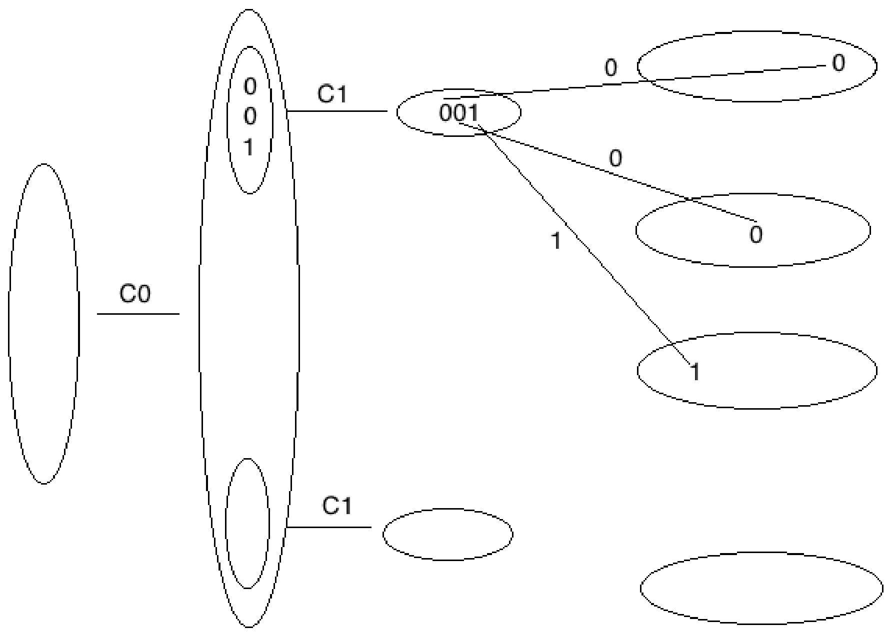
AEL code has 3 steps for encoding a) Use C_0 the encode message $\Sigma^k \to \Sigma^{nl}$ C_0 has rate $1-\epsilon$ and relative distance ϵ b) For each block of size l, use small code C_1 to encode $\Sigma^l \to \Sigma^d$. C_1 has rate l/d relative distance δ c) Use Bipartite graph B of d regular and n vertexes for both side to split Σ^d into d different blocks and concatenate each block on the right hand side. The Formal construction of AEL code is

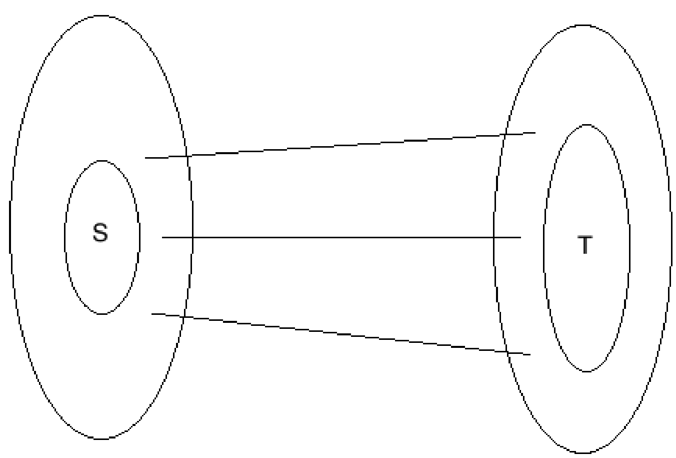
1. Start with message in Σ^k, encode this message to Σ^{nl} by C_0
2. Assign each of the l elements to left vertex of expander graph G.
3. Encode each element using C_1 from Σ^l to Σ^d for every left vertex
4. Place one of these d elements on each edge leaving the vertex.
5. Each right vertex is assigned to d-tuple corresponding to edges incident to it.

In total the encoding is $\Sigma^k \to (\Sigma^d)n$. So the rate is $\frac{k}{dn}$. To get good distance, we need stronger assumption of the bipartite graph B rather than just (α, δ) expansion since several changes on left can collide on one change of the right set.

Let's see what happens if we use a random bipartite graph.

Let S on left be vertexes that are non-zero. Let T on right be vertexes that are non-zero. Let $\Gamma(i)$ be the neighbor set of vertex i. For random bipartite graph B and typical vertex i on left, $|\Gamma(i) \cap T| \sim \frac{|T|}{n}d$. And for $i \in S$, since the distance of C_1 is δ, we know that at least δd coordinates are non-zero. So we get


C0
C1
0
0
1
001
0
0
0
1
0
1
C1


S
T

$|\Gamma(i) \cap T| \geq \delta d$. So if $\frac{|T|}{n} < \delta$, then S is not a typical set which means $|S|$ is small. This gives the following definitions.

Definition 3 ((δ, ϵ) Sampler). *Let $\Gamma_\delta(T)$ denote the set $\{i$ in left set such that $|\Gamma(i) \cap T| \geq \delta d\}$. Then B is (δ, ϵ)-sampler if for all T as a subset of right set, $|T| \leq \delta\epsilon n$, we have $|\Gamma_\delta(T)| < \epsilon n$*

We will use bipartite graph B as (δ, ϵ) Sampler. The existence of such graph is guaranteed by following theorem

Theorem 4. *For all $\delta, \epsilon > 0$, $\exists d$ such that for large n, there exists d-regular bipartite graphs on n-vertexes which are (δ, ϵ)-samplers.*

Directly using the definition of (δ, ϵ)-samplers, we can prove the following theorem

Theorem 5. *With encoding C_0 of rate R_1 and relative distance ϵ and encoding $C_!$ of rate R and relative distance δ, conduct the AEL encoding with (δ, ϵ) sampler B, we can obtain final code C_f of rate $R_1 R$ and relative distance $\delta\epsilon$*

Proof. Proof by contradiction. Suppose the relative distance is less than $\delta\epsilon$, then there is T as the set of non-zero elements on the right which has $|T| \leq (\delta, \epsilon)n$. By the definition of (δ, ϵ) sampler, if we pick S as the set of non-zero elements on the left, we have $|\Gamma_S(T)| \leq \epsilon n$ which contradicts the relative distance assumption of C_0 and C_1 □

Now we can apply above theorem. If we plug in C_0 of rate $1 - 1 - O(\epsilon)$ and distance ϵ, we get C_f as a long code of rate $(1 - O(\epsilon)R$ and distance $\delta\epsilon$ which is near the Singleton bound. Also the alphabet of C_f is Σ^d which is a constant. If we further assume C_0 is linear time decodable from ϵ fraction of error and has rate $1 - O(\epsilon)$ and C_1 is a good error correcting code such that $R = 1 - \delta$. Then C_f has rate $1 - \delta - O(\epsilon)$ and C_f is correctable in linear time from $\frac{\delta}{2} - \epsilon$ fraction of errors. Next we will talk about decoding algorithm for AEL code.

4 Guruswami-Indyk algorithm

The natural way to decode AEL codes is to reverse the steps of the encoding procedure. That is, given an output message, we can travel backwards on the edges of B to get candidate codewords for each left vertex .Then, we use a decoding algorithm for the code C_1 to get the message associated with each left vertex. Once we have these values, we can use a decoding algorithm for C_0 to get the original message. The formal description of the algorithm is

1. Traverse along the edges from the right vertex to its d neighbors.
2. Using the edge weights form the codeword for each of vertex on the left side.
3. Apply decoding algorithm of C_1 to get the initial left vertexes.
4. Apply decoding algorithm of C_0 to get the initial message sent.

Also we can generalize this algorithm to list-decoding. Here are several typical applications

Example 1. *Let C_0 be linear time decodable codes from $\Omega(\epsilon)$ fraction of errors. Let C_1 be good decodable codes of rate $1 - \delta$ and decodable from δ fraction of errors. Let B be $(\delta/2, \epsilon')$ samplers. Then the combined code is linear time decodable from $\delta/2 - \epsilon$ fraction of errors.*

Example 2 (Guruswami-Rudra). *Let C_0 be list recoverable codes from $R + \epsilon n$ agreement.Let C_1 be good codes of rate $1 - \delta$ and decodable from δ fraction of errors. Let B be (δ, ϵ) samplers. Then the combined code is polynomial time list-decodable from $\delta - \epsilon$ fraction of errors.*

Example 3. *Let C_0 be linear time list decodable codes from $\Omega(\epsilon)$ fraction errors. Let C_1 be good $1-\epsilon'$ list decodable codes. Let B be (δ, ϵ') samplers. Then the combined code is linear time list-decodable from $1-\epsilon''$ fraction of errors.*

We briefly mention the notion of list-recoverable. Here is the definition of list recoverable problem

Definition 6 (list recovery problem)**.** *Given code $C_0 \subset \Sigma^n$, given set $S_1, \ldots, S_n \subset \Sigma$ where $|S_i| \leq t$. The problem is to find all codewords $w \in C_0$ such that $|\{i|w_i \in S_i\} \leq \epsilon n$*