Lecture 7: Distance amplification and list decodable codes

Error-Correcting Codes (Spring 2016)
Rutgers University
Swastik Kopparty
Scribe: Mrinal Kumar

Plan for the lecture

The plan for this lecture is two fold:

- In the first half, we will look at another application of expander graphs to distance amplification. Recall that in the last lecture, we looked at some such constructions.
- In the second half of the class, we will start with the notion of list decoding and list decodable codes. This theme will continue in the next class as well.

1 Distance amplification via expanders

So far, we have seen three versions of application of bipartitie expanders to construction of error correcting codes.

- We saw a code defined via parity checks on the neighbors of a vertex.
- We generalized this, by additionally asking the neighbors to be the code words of an inner code.
- We also saw the *expander symbol pushing* based construction of Alon et al.

The construction we are going to see is by Alon, Edmonds and Luby. The main gadget to be used will again be an appropriate notion of a bipartitie expander. We define the notion and state some of its properties.

1.1 γ-Expanders

Definition 1. *Let $G(L, R, E)$ be a d-regular bipartitie graph, with bipartitions L and R and the set of edges being E. Let $|L| = |R| = n$. For a paramter $\gamma \in (0, 1)$, G is said to be a γ-expander if $\forall X \subseteq L$ and $\forall Y \subseteq R$,*

$$\left| E(X, Y) - \frac{d}{n} \cdot |X| \cdot |Y| \right| \leq \gamma dn$$

Here, $E(X, Y)$ denotes the number of edges between vertices in X and vertices in Y.

Observe that if G was a truly random d-regular bipartitite graph with n vertices on each side, then the number of edges between two sets $X \subseteq L$ and $Y \subseteq R$ is about $\frac{d}{n} \cdot |X| \cdot |Y|$ in expectation. Definition 1 says that for a γ-expander, the actual number of edges between X and Y is within an error of γdn of this expectation. Clearly, if the size of the sets X and Y is small (say much smaller than $\sqrt{n}$), then this condition is not saying anything non-trivial, but for sets of larger size, this is a non-trivial requirement.

Typically, we will think of the degree d being small. Moreover, we will identify the sets L and R with $\{v_1, v_2, \ldots, v_n\}$ and the set R with $\{u_1, u_2, \ldots, u_n\}$.

We now state the following useful fact about existence of such expanders.

Fact 2. *A random d-regular bipartitite graph is a $O(\frac{1}{\sqrt{d}})$-expander. This implies that for every constant $\gamma \in (0, 1)$, we can choose a large enough d, such that for all sufficiently large n, there is a γ-expander $G(L, R, E)$ which is d-regular and $|L| = |R| = n$. To choose a random d-regular bipartitite graph, we pick d pefect matchings independently and uniformly at random from a $K_{n,n}$, and take the union of edges, counting with multiplicity.*

In fact, we know of efficient explicit construction of such expanders. For our application to distance amplification, this would be important.

The following property of γ-expanders, which esentially follows from Definition 1 would be useful for us.

Corollary 3. *Let $G(L, R, E)$ be a d-regular γ-expander with $|L| = |R| = n$. Let β and $\epsilon > 0$ be parameters, and let $Y \subseteq R$ be an arbitrary subset of R of size βn. Then,*

$$\left|\left\{v \in L : |E(\{v\}, Y)| \geq (\beta + \epsilon)d\right\}\right| \leq \frac{\gamma}{\epsilon} n$$

Proof. Let

$$X = \left\{v \in L : |E(\{v\}, Y)| \geq (\beta + \epsilon)d\right\}$$

We know that $E(X, Y) \geq (\beta + \epsilon)d|X|$.

So, by Definition 1, we get

$$\left|(\beta + \epsilon)d|X| - \frac{d}{n}|X||Y|\right| \leq \gamma dn$$

Substituting $|Y| = \beta n$, we get

$$\left|(\beta + \epsilon)d|X| - \frac{d}{n}|X| \cdot \beta n\right| \leq \gamma dn$$

This gives,

$$|X| \leq \frac{\gamma}{\epsilon} n$$

□

1.2 AEL distance amplification

We will now discuss the application of γ-expanders to distance amplification for error correcting codes. We follow the following notation.

- $\mathcal{C} \subseteq \Sigma^n$ is the outer code of block length n, relative distance Δ and rate R_0.
- $\Phi : \Sigma \to \Sigma_0^d$ is a one to one map from the alphabet Σ to strings of length d over a smaller alphabet Σ_0. We will think of d as being small, and the image of Σ under Φ to be itself an error correcting code of relative distance δ and rate R_1. We call this the inner code, and abuse notation to refer to it as Φ.
- Let $\Psi : \Sigma_0^d \to \tilde{\Sigma}$ be a bijective map from strings of length d over Σ_0 to a new larger alphabet $\tilde{\Sigma}$. Unlike Φ, we do not need any additional properties from Φ apart from it being a bijection.
- Let $G(L, R, E)$ be a given d-regular γ-expander. As stated, we will identify the sets L with $\{v_1, v_2, \ldots, v_n\}$ and R with $\{u_1, u_2, \ldots, u_n\}$. For every vertex $v \in L$ (and similarly in R), by $N(v) = (u_{i_1}, u_{i_2}, \ldots, u_{i_d})$ where $i_1, i_2, \ldots, i_d \in [n]$ and $i_1 \leq i_2 \leq \cdots \leq i_d$, we denote the neighbors of v ordered in a canonical way.

Our goal is to construct a new code with block length n and possibly[1] distance better than Δ while not loosing too much in the rate. We do this construction by showing how to map every codeword $\overline{c} = (c_1, c_2, \ldots, c_n) \in \mathcal{C}$ to a codeword of the new code. And, then we study the properties of the new code constructed.

1. We concatenate $\overline{c}$ with Φ to get a new word $\Phi(\overline{c})$ in $(\Sigma_0^d)^n$. $\Phi(\overline{c})$ is defined as
$$\Phi(\overline{c}) = (\Phi(c_1), \Phi(c_2), \ldots, \Phi(c_n))$$
2. For every $i \in [n]$, let $N(v_i) = (u_{i_1}, u_{i_2}, \ldots, u_{i_d})$ be the ordered neighborhood of vertex v_i. Let $\Phi(c_i) = (a_1, a_2, \ldots, a_d)$. Then for every $j \in [d]$, we label the edge (v_i, u_{i_j}) in G by the alphabet a_j. We now have a labelling of every edge in G.
3. We now collect the edge labels to label the vertices in R first by strings in Σ_0^d in the natural way. And, then we map these labels using Ψ to get labels in $\tilde{\Sigma}$. More precisely, for every $i \in [n]$, let $N(u_i) = (v_{i_1}, v_{i_2}, \ldots, v_{i_d})$ be the ordered tuple of neighbors of u_i. And, for every $j \in [d]$, let $a_j \in \Sigma_0$ be the label of the edge (v_{i_j}, u_i) as defined in step two. Then, the vertex u_i is labelled by the string $\Psi(a_1, a_2, \ldots, a_d) \in \tilde{\Sigma}$. Let the label of u_i be denoted by c'_i.
4. The codeword $\overline{c} = (c_1, c_2, \ldots, c_n)$ is mapped to $\overline{c}' = (c'_1, c'_2, \ldots, c'_n)$.

The new code $\mathcal{C}' \subseteq \tilde{\Sigma}^n$ is defined as
$$\mathcal{C}' = \left\{ (c'_1, c'_2, \ldots, c'_n) : (c_1, c_2, \ldots, c_n) \in \mathcal{C} \right\}$$

We now try to understand the parameters of the code $\mathcal{C}'$.

[1] The final distance depends on the parameters, but the goal is to amplify distance of $\mathcal{C}$.

Rate of $\mathcal{C}'$ Observe that the rate of $\mathcal{C}'$ equals the rate of code obtained by concatenating the outer code $\mathcal{C}$ with the inner code Φ. So, it is $R_0 \cdot R_1$.

Observe that the above construction is a genralization of the construction of ABNNR that we saw in the last lecture, since in that construction, the inner code is always the repetition code. So, R_1 is far from one. Here, on the other hand, we can pick R_1 to be much closer to 1. So, if we care about keeping the rate of the codes we are working with close to 1, then the above transformation is preferable to the ABNNR construction.

Distance of $\mathcal{C}'$ Let a' and b' be two distinct codewords in $\mathcal{C}'$ and let a and b be their preimages in $\mathcal{C}$. Let the set of coordinates where a' and b' be the set $Y \subseteq [n]$. Similarly, let the set of coordinates where a and b differ be denoted by Z. Our goal is to lower bound the size of $|Y|$.

Let G_a and G_b be the labellings of the graph G under a and b respectively, as defined in the construction of $\mathcal{C}'$. Not surprisingly, we will analyze the distance between a' and b' by looking at G_a, G_b and the original graph G. We make the following simple observations.

Observation 4. *If a and b differ on the coordinate $i \in [n]$, then at least δd edges incident to the vertex v_i in G are labelled differently in G_a and G_b.*

Proof. Let a_i and b_i be the symbols in the i^{th} coordinates of a and b respectively. By our hypothesis, $a_i \neq b_i$. This implies that $\Phi(a_i)$ and $\Phi(b_i)$ differ on at least δd coordinates, since the image of Φ is an error correcting code of block length d and distance δd. From the construction of $\mathcal{C}'$, the edges coming out of vertex v_i are labelled using $\Phi(a_i)$ in G_a (by ordering the edges canonically), and similarly for G_b. Hence, they differ on at least δd locations. □

Observation 5. *Let (v_i, u_j) be an edge in G which is labelled differently in G_a and G_b. Then, the codewords a' and b' differ on their j^{th} coordinate.*

Proof. From the construction of $\mathcal{C}'$, it follows that the symbol on the j^{th} coordinate of a' is obtained by taking the ordered tuple of labels of the edges incident to u_j and writing it as a symbol of the larger alphabet $\tilde{\Sigma}$. Since G_a and G_b differ on the label of an edge incident to the vertex u_j, it follows that the resulting codewords a' and b' will differ on their j^{th} coordinate. □

Together, the two observations above tell us that every vertex in the set Z has at least δd many neighbors in the set Y. We would like to conclude from this, the set Y is large. Intuitively, the strategy is the following: if the codewords a' and b' were not far apart i.e the set Y was too small, then Z must also be small, and for an appropriate choice of parameters, this would imply that the codewords a and b must also be close to each other, which would contradict that $\mathcal{C}$ is a code with good distance.

We now make this intuition formal.

Lemma 6.

$$|Y| > n\left(\delta - \frac{\gamma}{\Delta}\right)$$

Proof. Let $\epsilon = (\delta - \frac{|Y|}{n})$. As we discussed, every vertex in the set Z in G has at least $\delta d = (\frac{|Y|}{n} + \epsilon)$ many neighbors in the set Y. Therefore, by Corollary 3, it follows that

$$|Z| \leq \frac{\gamma}{\epsilon} n$$

But, Z is the set of coordinates where the codewords a and b of $\mathcal{C}$ differ, and hence $|Z| \geq \Delta n$. Putting the two inequalities together, we get

$$|Y| > n\left(\delta - \frac{\gamma}{\Delta}\right)$$

□

Choosing parameters We will pick the parameters such that $\frac{\gamma}{\Delta}$ is much much smaller compared to δ. For this, we pick γ to be very small and Δ to be slightly larger than γ but still very small. Observe that in this case, the relative distance of $\mathcal{C}'$ is very close to the relative distance δ of the inner code. Also, since we can choose Δ to be extremely small (but large compared to γ), we can pick our outer code $\mathcal{C}$ to have rate close to 1. In this case, the rate of $\mathcal{C}'$ is roughly the rate of the inner code given by Φ.

Therefore, we get rate and distance close to that of the inner code Φ, although now, we have a much larger block length. Of course, there is the cost of working over a much larger alphabet $\tilde{\Sigma}$. In particular, if we pick the inner code to be the Reed-Solomon code with constant block length, and hence constant sized alphabet, and the outer code $\mathcal{C}$ to be a code with rate close to 1, then we get a code with longer block length, with large (but still constant) sized alphabet, with distance $\delta - \epsilon$ and rate $1 - \delta - \epsilon'$, i.e getting close to the Singleton bound. This is in contrast to Reed-Solomon codes of large block length, where the alphabet size grows with block length.

2 List decodable codes

Let $\mathcal{C} \subseteq \Sigma^n$ be a code with relative distance δ. It is clear that for every $x \in \Sigma^n$, $\left|B(x, \frac{\delta}{2}) \cap \mathcal{C}\right| \leq 1$. Here, $B(x, \alpha)$ denotes the ball of radius αn centered at x.

Finding this intersection is the unique decoding question that we have been studying so far. In the list decoding setting, we consider a ball of larger radius ρ and the goal is to find the intersection $B(x, \rho) \cap \mathcal{C}$. Since, $\rho > \frac{\delta}{2}$, the size of the intersection could be larger than one and the goal is to output the entire list. Hence, the name *list decoding*.

List decoding, in addition to being a fairly natural generalization of unique decoding, and being a natural notion to study on its own has found surprising applications in complexity theory and to a better understanding of unique decoding itself.

We will use the parameters L and ρ to index the list decoding problem in the following natural way. ρ denotes the radius of the ball for which we will be interested in finding the intersection $B(x, \rho) \cap \mathcal{C}$, and L will be the largest value that the size of such an intersection can take. We will refer to such a code as a (ρ, L)-list decodable code.

For $\rho = \frac{\delta}{2}$ and $L = 1$, this is the classical unique decoding problem that we have seen so far. For larger values of ρ and L, we will be interested in the following questions:

- Find the code $\mathcal{C}$ of the highest rate such that $\left|B(x, \rho) \cap \mathcal{C}\right| \leq L$. Prove upper, lower bounds on the size/rate that such a code can achieve.
- Construct such a code, and find efficient decoding, encoding algorithm for the code.

Surprisingly, in some sense, which we will make precise as we go along, our understanding of these problems for larger values of L is better than our understanding of the $L = 1$ case.

2.1 Upper and lower bounds on the rate

For simplicity, we will take $\Sigma = \{0, 1\}$ for this discussion.

The following theorem gives us an upper bound on the rate of (ρ, L)-list decodable codes.

Theorem 7. *Let $\mathcal{C} \subseteq \{0,1\}^n$ be a (ρ, L)-list decodable code with rate R. Then,*

$$\mathit{Rate}(\mathcal{C}) \leq 1 - H(\rho) + \frac{\log L}{n}$$

Proof. The volume packing argument immediately gives us an upper bound on the rate of a (ρ, L)-list decodable code. If we draw a ball of radius ρn around every point in $\mathcal{C}$, then each point in $\{0,1\}^n$ is covered with multiplicity at most L. Therefore, we have

$$\left|B(\rho)\right| \cdot \left|\mathcal{C}\right| \leq L \cdot 2^n$$

Here, we are abusing notation and using $B(\rho)$ to denote the size of the hamming ball of radius ρn. Plugging in the standard estimates for $B(\rho)$, this gives,

$$\text{Rate}(\mathcal{C}) \leq 1 - H(\rho) + \frac{\log L}{n}$$

□

We will now use the probabilistic method to show a lower bound on the rate of a (ρ, L)-list decodable code. To this end, we prove the following theorem.

Theorem 8. *For every choice of parameter n, which is sufficiently large and for every choice of R, ρ, L such that*

$$R + H(\rho) + 1/(L+1) < 1$$

there exists a code $\mathcal{C} \subseteq \{0,1\}^n$ of rate R which is (ρ, L)-list decodable.

Proof. We will construct a code of rate R which we will fix at the end to make the calculations work. Let us construct a set $\mathcal{C}$ by picking 2^{Rn} strings from $\{0,1\}^n$, independently and uniformly

at random. We would now try to understand, the conditions under which such a set will be a (ρ, L)-list decodable code. Clearly, $\mathcal{C}$ is not a (ρ, L)-list decodable code if and only if there exists an $x \in \{0,1\}^n$ such that the ball of radius ρn around x contains more than L points from $\mathcal{C}$. We call any such x to be 'bad'.

Let us fix $x \in \{0,1\}^n$. Then,

$$\Pr\left[\left|B(x,\rho) \cap \mathcal{C}\right| > L\right] \leq \binom{2^{Rn}}{L+1} \cdot \left(\frac{B(\rho)}{2^n}\right)^{L+1}$$

Thefore, by a union bound, the probability that there is a bad x, and hence $\mathcal{C}$ is not (ρ, L)-list decodable is at most

$$\Pr\left[\exists x \text{ such that } \left|B(x,\rho) \cap \mathcal{C}\right| > L\right] \leq 2^n \cdot \binom{2^{Rn}}{L+1} \cdot \left(\frac{B(\rho)}{2^n}\right)^{L+1}$$

Since L is small compared to 2^n, by simplifying, we get

$$\Pr\left[\exists x \text{ such that } \left|B(x,\rho) \cap \mathcal{C}\right| > L\right] \leq 2^{n(L+1)(R+H(\rho)-1+1/(L+1))}$$

Now, observe that for R such that $R + H(\rho) < 1$ and sufficiently large L, this probability is $\exp(-\Omega(n))$. Therefore, for $R < 1 - H(\rho)$, there exists some large enough L, such that there is a (ρ, L)-list decodable code of rate R. □

Note that this is in contrast to the case of unique decoding, where the upper bound for the rate looks like $1 - H(\rho)$, while the lower bound is $1 - H(2\rho)$. Hence, in this sense, things are better for list decodable codes with larger values of L.

Explicit constructions The problem of explicitly constructing list decodable codes matching the above probabilistic argument is open over small alphabets. However, over alphabets of large size such explicit constructions are known. The first such construction was by Guruswamy and Rudra in 2006, and now other constructions are known.

In this lecture and the next, we will address the following aspects of list decodable codes:

- Johnson Bounds - these are statements which show that if a code has distance at least δ, then it is also list decodable for some $\rho > \delta/2$ and $L > 1$.
- We will then see some constructions of list decodable codes.
- We will then see some list decoding algorithms for list decodable codes.

Johnson bound

We start with the following version of the Johnson bound.

Lemma 9 (Johnson bounds - version 1)**.** *Let $\mathcal{C} \subseteq \Sigma^n$ be a code with distance at least δ. Then, $\mathcal{C}$ is also (ρ, L)-list decodable with $\rho = 1 - \sqrt{1-\delta}$ and $L \leq n$.*

Proof. Before we proceed with the proof, note that ρ is always grater than $\delta/2$, which makes sense since otherwise, the problem is the same as that of unique decoding.

Let $x \in \Sigma^n$ be an arbitrary string of length n over alphabet Σ. To prove the lemma, we would have to show that the number of codewords which disagree with x on at most ρn coordinates is at most n. The proof is elementary but clever. We will reduce this problem to studying some extremal properties of a bipartitie graph. The proof is similar to the proof of the bipartite version of Turan's theorem for graphs.

Let $c_1, c_2, \ldots, c_L$ be all codewords in $\mathcal{C}$ which disagree with x on at most ρn coordinates. Construct a bipartite graph $G(A, B, E)$ with

- $A = \{c_1, c_2, \ldots, c_L\}$
- $B = [n]$
- For $c_i \in A$ and $j \in B$, there is an edge between c_i and j if c_i and x agree on the j^{th} coordinate.

The following properties follow from the definitions:

- The degree of every vertex in A in G is at least $(1 - \rho)n$.
- For any two distinct vertices $c_i, c_j \in A$, the number of common neighbors of c_i and c_j is at most $(1-\delta)n-1$. This follows from the fact that c_i and c_j are codewords of a code of distance at least δ.

Now, we prove upper and lower bounds on the number of "vees" in the graph G, i.e number of sets $\{c_i, c_j, k\}$, such that $c_i, c_j \in A$, $k \in B$, $(c_i, k) \in E$ and $(c_j, k) \in E$. Clearly, the number of vees is at most the number of distinct pairs $\{c_i, c_j\}$ multiplied by the maximum number of common neighbors any such pair can have in G. This is at most $\binom{L}{2} \cdot ((1 - \delta)n - 1)$.

For a lower bound on the number of vees, observe that for any fixed vertex $j \in B$, the number of vees that j is a part of is equal to $\binom{d_j}{2}$ where, d_j is the degree of j in G. So, the total number of vees is

$$\sum_{i \in B} \binom{d_i}{2}$$

Now, observe that $\sum_{i \in [n]} d_i$ is the number of edges in G, and hence is at least $(1 - \rho)nL$. Also, the shape of the function $f(z) = z(z - 1)/2$ is convex i.e the secant line between any two points lies above the curve of the function. Therefore,

$$1/n \cdot \sum_i \binom{d_i}{2} \geq \binom{\sum_i d_i/n}{2}$$

So, the number of vees is at least

$$1/2 \cdot \Big(\sum_i d_i\Big) \cdot \Big(\sum_i d_i/n - 1\Big)$$

Comparing the upper and lower bounds, and using the fact that $\sum_i d_i \geq (1-\rho)nL$, we get

$$\binom{L}{2} \cdot \Big((1-\delta)n - 1\Big) \geq 1/2 \cdot \Big(1-\rho\Big) \cdot nL \cdot \Big((1-\rho)L - 1\Big)$$

Simplifying, we get

$$(L-1) \cdot \Big((1-\delta)n - 1\Big) \geq \Big(1-\rho\Big)^2 \cdot nL - \Big(1-\rho\Big) \cdot n$$

Now, from the choice of ρ, we have $(1-\rho)^2 = 1-\delta$. Substituting this back, and simplifying, we get

$$L \leq \rho(1-\rho)n + 1$$

This completes the proof of the lemma. □