Lecture 4: Expander Codes

Lecturer: He Sun

In this lecture we discuss the applications of expander graphs to error-correcting codes. Let us start with some preliminaries on coding theory.

1 Preliminaries

Suppose that Alice wants to send Bob a k-bit message over a noisy channel (i. e., the channel flips some bits of the message). To make sure that Bob can get the correct message, Alice instead of sending the original message, encodes this k-bit message into n-bit encoding, such that Bob can recover the message from the channel if the codeword was not corrupted too bad by the channel.

Definition 4.1 *A q-ary code is a set $\mathcal{C} \subseteq \Sigma^n$, where Σ is an alphabet of size q. Elements of $\mathcal{C}$ are called codewords. Some basic parameters:*

1. *n is the block length.*
2. *$k = \log_2 |\mathcal{C}|$ is the message length.*
3. *$\rho = k/n$ is the relative rate of the codes.*

An encoding function for $\mathcal{C}$ is an injective mapping $\text{Enc} : \{0,1\}^k \to \{0,1\}^n$. Given such an encoding function, we view the string in $\{0,1\}^n$ as messages. The code is explicit if Enc is computable in polynomial time.

Definition 4.2 *For any two strings $x, y \in \{0,1\}^n$, the Hamming distance of x and y, denoted by $\Delta(x,y)$, is the number of bits where they differ. The Hamming distance of a code $\mathcal{C}$ is $d(\mathcal{C}) = \min_{x,y\in\mathcal{C}, x\neq y} \Delta(x,y)$. The relative distance of $\mathcal{C}$ is defined by $\delta = d(\mathcal{C})/n$.*

We refer to a code $\mathcal{C}$ that maps k-bit messages to n-bit codewords with distance d as a (n,k,d)-code.

Definition 4.3 *A code $\mathcal{C}$ is linear if $\mathcal{C}$ satisfies the following properties:*

1. *$0^n \in \mathcal{C}$.*
2. *If $x, y \in \mathcal{C}$, then $x \oplus y \in \mathcal{C}$.*

We use $[n,k,d]$-code to express a linear code that maps k-bit messages to n-bit codewords with distance d.

2 Expander Codes

For any d-regular n-vertex graph $G = (V, E)$, the *double cover* of G is a bipartite graph $G_B = (L \cup R, E')$, where $L = \{\ell_1, \cdots, \ell_n\}, R = \{r_1, \cdots, r_n\}$ and $(u, v) \in E$ if and only if (ℓ_u, r_v) and (ℓ_v, r_u) are in E'.

Let $\mathcal{C}$ be a $[d, rd, \delta d]$-code. Given a code $\mathcal{C}$ and a d-regular n-vertex graph G, we construct $\mathcal{C}' = Z(\mathcal{C}, G)$ as follows. The length of the codewords in $\mathcal{C}'$ is dn. We give the edges of G_B an arbitrary labeling and use $e_1, \cdots, e_{dn}$ to express the edges in G'. Moreover, we consider each codeword $x \in \{0,1\}^{dn}$ as an assignment of $e_1, \cdots, e_{dn}$ and let the assignment of e_i be x_i. Given these, a string $x \in \{0,1\}^{dn}$ is in $\mathcal{C}'$ if and only if for every $v \in L \cup R$ with adjacent edges $e_{i_1} \cdots e_{i_d}$, $x_{i_1} \cdots x_{i_d}$ is a codeword in $\mathcal{C}$.

Lemma 4.4 *For any linear code $\mathcal{C}$, the distance of $\mathcal{C}$ equals the minimum Hamming weight of a non-zero codeword.*

Theorem 4.5 *Suppose $\mathcal{C}$ is a $[d, rd, \delta d]$-code with rate $r < 1/2$ and G is a d-regular expander on n vertices with spectral expansion λ. Then $\mathcal{C}' = Z(\mathcal{C}, G)$ is a $[dn, (2r-1)dn, \delta(\delta-\lambda)dn]$-code.*

Proof: By construction, it is easy to see that the length of the codewords in $\mathcal{C}'$ is dn. Since $\mathcal{C}$ is of the form $[d, rd, \delta d]$, then the number of constraints for $\mathcal{C}$ is $d - rd = (1-r)d$. Because, for constructing $\mathcal{C}'$, such $(1-r)d$ constraints are applied for each vertex in G_B, so the number of constraints for $\mathcal{C}'$ is at most

$$2n(1-r)d = nd(2-2r) = nd(1-(2r-1))$$

and the rate of $\mathcal{C}'$ is at least $(2r-1)$.

Now we analyze the rate of $\mathcal{C}'$. Let x be a codeword in $\mathcal{C}'$. Define $X = \{e_i | x_i = 1\}$. Let S and T be the respective sets of left and right vertices that are adjacent to one edge in X. By Lemma 4.4, we get $|X| \geq \frac{\delta d}{2}(|S| + |T|)$.

On the other side, by the Expander Mixing Lemma, wet get

$$|X| \leq |E(S,T)| \leq \frac{d \cdot |S||T|}{n} + \lambda d\sqrt{|S||T|}.$$

Thus

$$\frac{\delta d}{2}(|S| + |T|) \leq \frac{|S||T|}{n} + \lambda\sqrt{|S||T|}.$$

Since $|S||T| \leq \frac{(|S|+|T|)^2}{4}$, we get

$$\frac{\delta}{2}(|S| + |T|) \leq \frac{(|S| + |T|)^2}{4n} + \frac{\lambda}{2}(|S| + |T|),$$

which implies $|S| + |T| \geq 2n(\delta - \lambda)$. Therefore the distance of $\mathcal{C}'$ is at least

$$|X| \geq \frac{\delta d}{2} \cdot 2n(\delta - \lambda) = \delta(\delta - \lambda)dn.$$

■

```
V_0 ← L
i ← 0;
while ∃v such that x_v ∉ C do
    foreach v ∈ V_i such that x_v ∉ C do
        decode x_v to the nearest codeword in C.
    end
    if V_i = L then
        V_{i+1} ← R
    else
        V_{i+1} ← L
    end
    i ← i + 1
end
return x
```

Algorithm 1: Decoding Algorithm

3 Decoding Algorithm

Theorem 4.6 *Suppose $\mathcal{C}$ is a $[d, rd, \delta d]$-code and G is a d-regular expander on n vertices with spectral expansion $\lambda < \delta/3$. Then for all $\alpha, 0 \leq \alpha \leq 1$, there is a decoding algorithm to correct $\alpha\frac{\delta}{2}\left(\frac{\delta}{2} - \lambda\right) dn$ errors in $O\left(\frac{\lg n}{\lg(2-\alpha)}\right)$ time.*

Proof: Because $\mathcal{C}'$ is linear, we assume that the correct codeword to the corrupted codeword x is 0^n. Let $x^{(0)}$ be the initial corrupt codewords and after i rounds, let the working word be $x^{(i)}$.

Define $E^{(i)} = \left\{e | x_e^{(i)} = 1\right\}$. Let $S^{(i)}$ be the set of vertices in V_{i-1} with edges in $E^{(i)}$. Consider the set $E\left(S^{(i)}, S^{(i+1)}\right)$. Because $S^{(i+1)}$ is the set of vertices that do not correctly decode the zero words in the $(i+1)$-th round, therefore with respect to $E^{(i)}$, the degree of each vertex in $S^{(i+1)}$ is at least $\frac{\delta d}{2}$. We have

$$\left|E\left(S^{(i)}, S^{(i+1)}\right)\right| \geq \frac{\delta d}{2}\left|S^{(i+1)}\right|. \tag{1}$$

By the Expander Mixing Lemma,

$$\left|E\left(S^{(i)}, S^{(i+1)}\right)\right| \leq \frac{d\left|S^{(i)}\right| \cdot \left|S^{(i+1)}\right|}{n} + \lambda d\sqrt{\left|S^{(i)}\right| \cdot \left|S^{(i+1)}\right|}. \tag{2}$$

On the other side, we know that $\left|E^{(0)}\right| \leq \alpha \cdot \frac{\delta}{2}\left(\frac{\delta}{2} - \lambda\right) dn$ and $\left|E^{(0)}\right| \geq \frac{\delta d}{2}\left|S^{(1)}\right|$. So

$$\left|S^{(1)}\right| \leq \alpha n\left(\frac{\delta}{2} - \lambda\right). \tag{3}$$

Combining Eq. (1) and Eq. (2), we get

$$\begin{aligned}
\frac{\delta \cdot d \cdot \left|S^{(i+1)}\right|}{2} &\leq \frac{d \cdot \left|S^{(i)}\right| \cdot \left|S^{(i+1)}\right|}{n} + \lambda d\sqrt{\left|S^{(i)}\right| \cdot \left|S^{(i+1)}\right|} \\
&\leq \frac{d \cdot \left|S^{(i)}\right| \cdot \left|S^{(i+1)}\right|}{n} + \frac{\lambda d}{2} \cdot \left(\left|S^{(i)}\right| + \left|S^{(i+1)}\right|\right) \\
&\leq \frac{d\left|S^{(i+1)}\right|}{n} \cdot \alpha n\left(\frac{\delta}{2} - \lambda\right) + \frac{\lambda d}{2} \cdot \left(\left|S^{(i)}\right| + \left|S^{(i+1)}\right|\right), \text{By Inequality (3)}
\end{aligned}$$

which implies

$$(\delta - \alpha(\delta - 2\lambda) - \lambda) \left|S^{(i+1)}\right| \leq \lambda \left|S^{(i)}\right|.$$

By the assumption that $\delta > 3\lambda$, we get

$$(2 - \alpha) \left|S^{(i+1)}\right| \leq \left|S^{(i)}\right|.$$

The formula above tells us the size of $S^{(i)}$ drops exponentially. So the algorithm will finish in $O\left(\frac{\lg n}{\lg(2-\alpha)}\right)$ rounds. ■