Coding Theory
Lecture 9

Lecture by Dr. Elette Boyle
Typeset by Steven Karas

2017-12-28
Last edited 18:22:54 2017-12-28

Disclaimer These lecture notes are based on the lecture for the course Coding Theory, taught by Dr. Elette Boyle at IDC Herzliyah in the fall semester of 2017/2018. Sections may be based on the lecture notes written by Dr. Elette Boyle.

Agenda January 11th's lecture has been rescheduled for Jan 2nd.

- Proof expander code decoding convergence
- Proof expander code decoding convergence
- Linear time encoding

1 Review

Last lecture we covered Binary Expander Codes, which are a family of codes that are defined as a mapping from a bipartite graph to a binary linear code.

Given a bipartite graph $G = (L, R, E)$ such that $|L| \geq |R|$. The adjacency matrix A_G of the bipartite graph is the parity check matrix of a $[|L|, \frac{|L|-|R|}{|L|}, ?]_2$ binary linear code.

The $(n, m, D, \gamma, \alpha)$ bipartite expander is a bipartite graph $G = (L, R, E)$ such that $|L| = n$, $|R| = m$, D-regular for which the following holds:

$$\forall S \subseteq L, |S| \leq \gamma n \Rightarrow |N(S)| \geq |S|\alpha$$

We also proved that there exists a $(n, m, D, \gamma, D(1-\varepsilon))$ bipartite expander where $D = \Theta\left(\frac{\log n/m}{\varepsilon}\right)$ and $\gamma = \Theta\left(\frac{\varepsilon m}{n}\right)$.

Let G be $(n, m, D, \gamma, D(1-\varepsilon))$ bipartite expander where $\varepsilon < 1/2$. Then, for any $S \subseteq L$, $|S| \leq \gamma n$, it follows that $|N(S)| \geq |S|D(1-\varepsilon)$ and $|U(S)| \geq |S|D(1-2\varepsilon)$.

We proved a weaker bound on the distance $d \geq \gamma n + 1$. The proof follows from contradiction that there is a unique neighbor for a codeword with γn 1s. But this contradicts the parity check, which means the minimum hamming weight is $\geq \gamma n + 1$.

Then we tightened this by proving that $d \geq 2\gamma(1-\varepsilon)n$. The proof follows from splitting the 1s of a codeword into two parts, one of size exactly γn. The overlapping and leftover neighbors shows this contradiction.

This gives us a binary linear code with rate $\frac{|L|-|R|}{|L|}$, which is asymptotically constant. The distance for this code is $\Theta(\frac{\varepsilon m}{D}(1-\varepsilon)) \geq \Theta(\frac{m}{\log n/m})$, which is a asymptotically constant relative distance.

2 LDPC Codes

Low density parity check codes. In 1962 Gallager proved[1] that random low-weight bipartite graph yields code with good rate and distance.

This was refined later to have the same properties with efficient decoding using expander graphs[2].

3 Efficient Decoding of Expander Codes

The intuition is given a bipartite graph and a codeword of vertices on the left side, we check the parity, and flip a bit in the codeword that has mostly unsatisfied parity checks (satisfied = 0, ...). It repeats this until convergence.

Formally Given a bipartite graph $G = (L, R, E)$ for an expander code $(n, m, D, \gamma, \alpha)$ and a received codeword $\vec{y}$. The decoding algorithm iteratively flips the bit y_l with a strict majority of unsatisfied parity checks. If each step, if such a $l \in L$ exists, then we flip the bit y_l. This makes all the parity checks swap satisfaction. This strictly decreases the total number of unsatisfied parity checks $r \in R$.

We will need to show that such a y_l exists until all parity checks are satisfied. We will also need to show correctness.

3.1 Decoding algorithm

Input $G = (\overset{n}{L}, \overset{m}{R}, E)$, $\vec{y} \in \{0,1\}^n$ codeword.

Setup For every $r \in R$, let $\text{Pcheck}(r) = \sum_{l \in N(r)} y_l$ over $\mathbb{Z}_2$. Initialize $\vec{y}'$ be $\vec{y}$, and $S_0, \ldots, S_D$ be $\emptyset$ sets. For every $l \in L$, let $j_l = |\{r \in N(l) : \text{Pcheck}(r) = 1\}| = \#$ neighbor constraints unsatisfied. Let $S_{j_l} \leftarrow S_{j_l} \cup \{l\}$.

Basically, put each left vertex in buckets according to how many parity checks are unsatisfied

Iterate Until $S_{\lceil \frac{D}{2} \rceil}, \ldots, S_D$ are empty, find the largest j such that $S_j \neq \emptyset$. Choose some $l \in S_j$. Flip the lth bit in $\vec{y}'$: $y'_l \leftarrow 1 - y'_l$. For each constraint $r \in R$ with $r \in N(l)$, update $\text{Pcheck}(r) \leftarrow 1 - \text{Pcheck}(r)$. For each $w \in N(r)$, update the number of satisfied constraints to $j_w \pm 1$. Move w to the new S_{j_w}.

Results If $S_0 = L$, then all the other sets are empty, and all the constraints are satisfied. In this case, output $\vec{y}'$. Otherwise, output failure.

3.2 Complexity analysis

Setup is $O(md)$ to compute $\text{Pcheck}(r)$. Initializing the buckets is $O(D)$. Assigning bits to buckets is $O(nD)$.

Iteration takes us $O(D)$ to find the bit to flip, $O(D)$ updates, which comprise $O(d)$ updates to sets and some constant factors.

Altogether, this gives us $O(md) + O(D) + O(nD)$ for setup, and $O(D) + O(dD)$ for each iteration. We perform at most m iterations, because each iteration we strictly reduce the number of unsatisfied constraints.

Overall, this gives us complexity of $O(ndD) + mO(Dd) = O(ndD)$.

3.3 Proof of correctness

Let G be a $(n, m, D, \gamma, \frac{3}{4}D)$ bipartite expander whose right degree is bounded by d. Note that D should be odd[1]. Recall that the distance of this code is $\geq 2\gamma(1-\varepsilon)n = \frac{3}{2}\gamma n$. This implies that the best decoding we can do is from $3/4\gamma n$.

Correctness follows from 2 claims: first, that the algorithm continues iterating until it is at a codeword, and that the resulting codeword is the desired one.

As a matter of notation, at any given iteration of the algorithm and current $\vec{y}'$, denote v as the number of errors (i.e. the distance of $\vec{y}$ to its closest codeword), and u as the number of unsatisfied constraints.

Claim 1 If current $\vec{y}'$ is such that $0 < v \leq \gamma n$, then there exists some $l \in L$ for which we can make progress by flipping it ($l \in S_j$ for some $j > D/2$).

Let $S \subseteq L$ be the set of positions in which $\vec{y}'$ disagrees with closest codeword $\vec{c}$. Note that $|S| = v$ and $v \neq 0 \Rightarrow S \neq \emptyset$.

Recall that G is an expander with expansion factor $\alpha = \frac{3}{4}D = (1 - \frac{1}{4})D$. Since $|S| \leq \gamma n$, it follows that $|U(S)| \geq (1 - 2\underset{\frac{1}{4}}{\varepsilon})D|S| = \frac{D}{2}|S|$ by our lemma about unique neighbors.

[1] This simplifies some of the analysis

Recall that $\vec{y}' = \vec{c} + \vec{e}$ where $\vec{c}$ is the codeword and $\vec{e}$ is the error vector. Note that $A_G\vec{y}' = A_G\vec{c} + A_G\vec{e} = \vec{0} + \vec{e} = \vec{e}$. From this linearity, it follows that any $r \in U(S)$ is necessarily an unsatisfied constraint (as exactly 1 neighbor is 1).

There are $D|S|$ total edges from S, and we know that $\geq \frac{D}{2}|S|$ of these edges go to unsatisfied constraints. By pigeonhole, there must exist some $l \in L$ such that at least $\frac{D}{2}$ of its D edges go to unsatisfied constraints.

Claim 2 Suppose that the original input $\vec{y}$ is such that $\Delta(\vec{y}, \vec{c}) \leq \frac{\gamma n}{2}$ for the codeword $\vec{c}$. We claim that for every iteration of the algorithm, it will always be the case that $\Delta(\vec{y}', \vec{c})$ is less than γn. Recall that the distance of $C(G)$ is at least $\frac{3}{2}\gamma n$.

Note that in each iteration, $\Delta(\vec{y}', \vec{c})$ changes by exactly 1 because we flip a single bit of $\vec{y}'$. If we ever reach a state where $\Delta(\vec{y}', \vec{c}) \geq \gamma n$, then we must reach a state where $\Delta(\vec{y}', \vec{c}) = \gamma n$. This cannot be the initial step, because our starting point is by definition $\Delta(\vec{y}, \vec{c}) \leq \frac{\gamma n}{2}$.

Suppose such a step exists. Consider the $\vec{y}'$ from this iteration, and let $S \subseteq L$ be the subset of indices such that $y'_l \neq c_l$. By the lemma, this means that $|S| = \gamma n \Rightarrow |U(S)| \geq (1 - 2\varepsilon)D|S| = \frac{1}{2}D\gamma n$. This implies that there are $\geq \frac{1}{2}D\gamma n$ unsatisfied constraints. However, we started with $\leq \gamma n/2$ errors, and as such the number of unsatisfied constraints was $\leq \frac{1}{2}D\gamma n$. Because we strictly decrease the number of unsatisfied constraints in each iteration of the algorithm, we cannot possibly arrive at such an intermediate step.

Together From claim 2, for any given iteration, we can use claim 1 to show that we will make progress.

4 Next Time

If you consider the bipartite graph with a singular R, it defines codewords for its neighbor set...

References

[1] Robert Gallager. Low-density parity-check codes. *IRE Transactions on information theory*, 8(1):21–28, 1962.

[2] Michael Sipser and Daniel A Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.