CS761 Derandomization and Pseudorandomness **2022-23 Sem I**

Lecture 6: 25-08-2022

Scribe: Amit Rajaraman *Lecturer: Rohit Gurjar*

For the next few lectures, we shall look at expander graphs and their applications. Expander graphs are interesting because they are "pseudorandom" – they behave like random objects.

We recall the subject of error correcting codes, pioneered by Shannon in 1948. It studies the idea of introducing "redundancy" when transmitting messages so that the messages are understandable even in the presence of errors.

Definition 6.1. *A* code $\mathcal{C}$ *is a subset of* $\{0,1\}^n$*. The elements of a code are called* codewords*.*

Definition 6.2. *Given* $x, y \in \{0,1\}^n$*, the* Hamming distance $d_H(x,y)$ *between* x *and* y *is* $|\{i \in [n] : x_i \neq y_i\}|$*. The distance* $d_H(\mathcal{C})$ *of a code* $\mathcal{C}$ *is* $\min_{\substack{x,y\in\mathcal{C}\\ x\neq y}} d_H(x,y)$*.*

The idea of coding theory is that given a word in $\{0,1\}^k$ (for some $k \leq n$), we translate it bijectively into a codeword in $\{0,1\}^n$ and transmit it to the receiver. Upon receiving the message, the received word is decoded in some way to get a word. This decoding involves first figuring out what codeword might have been transmitted, then using the inverse of the bijection to get back the original word in $\{0,1\}^k$.

One simple way is to decode a received word as the codeword closest to it, in the sense of the Hamming distance. This scheme allows the correction of errors if the received word is at Hamming distance less than $(1/2)d_H(\mathcal{C})$ from the transmitted word.

Definition 6.3. *The* rate *of a code is defined by*

$$\mathrm{Rate}(\mathcal{C}) = \frac{\log|\mathcal{C}|}{n}.$$

We also define the relative distance

$$\delta(\mathcal{C}) = \frac{d_H(\mathcal{C})}{n}.$$

One question that should immediately come to mind is: given a relative distance, what is the minimum rate required to achieve it? In less formal terms, what is the minimum amount of redundancy needed?

Problem. *Given constants* $\delta_0, r_0 \in (0,1)$*, when can we construct codes* $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$ *such that* $\delta(\mathcal{C}_n) \to \delta_0$ *and* $\mathrm{Rate}(\mathcal{C}_n) \to r_0$*?*

This also presents another follow-up question: if codes of the above form exist, do there exist efficient encoding and decoding algorithms for the code?

Consider another question.

Problem. *Suppose we have a randomized algorithm* $\mathcal{A}$ *with "one-sided error". This means that if* x *is in the language* L *of interest,* $\mathcal{A}(x)$ *is yes with probability* 1*, but if* x *is not in the language* L*,* $\mathcal{A}(x)$ *is no with probability* $\frac{15}{16}$*.*
How would one go about making the error probability very small, without using too many random bits?

One simple idea which we have discussed is to repeat the algorithm a large number of times and output no if we get a no at any point. Indeed, if we repeat it ℓ times, the error probability goes down to $\leq (1/16)^\ell$. However, the issue with this is that if the algorithm uses k independent random bits, then repeating it ℓ times requires ℓk independent random bits! Can we decrease this number to something like $\ell + k$? It turns out that this is possible.

The two questions we have described seem incredibly different, but the answers to both are yes, with the ideas behind both involving expander graphs.

Definition 6.4 (Magical graphs)**.** *A bipartite graph $G = (L \sqcup R, E)$ is said to be (n, m, d)-*magical*, $m \geq (3n/4)$, if*

1. *$|L| = n$, $|R| = m$,*
2. *G is d-regular, and*
3. *for every subset $S \subseteq L$ with $|S| \leq n/10d$, $|\Gamma(S)| \geq (5d/8)|S|$.*

Above, $\Gamma(S)$ denotes the neighbourhood of S.
Typically, n and m are of similar orders and d is a constant. This says that any "small" subset expands a lot – the neighbours of the vertices in the subset do not coincide too much. Ideally, with no intersection between neighbourhoods, we would have $|\Gamma(S)| = d|S|$, and we are demanding about half of this.
First, we shall see why magical graphs exist. Following this, we connect them to the questions we looked at earlier.

Theorem 6.5. *For $d \geq 24$ and sufficiently large n, (n, m, d)-magical graphs exist.*

Proof. For each vertex in L, choose its d neighbours randomly. Let $S \subseteq L$ with $|S| = s \leq n/10d$ and $T = R$ with $|T| = (5d/8)s$,

$$\Pr\left[\Gamma(S) \subseteq T\right] \leq \left(\frac{|T|}{m}\right)^{ds} \leq \left(\frac{5ds}{8m}\right)^{ds}.$$

This is for a *fixed* S, T however. Using the union bound,

$$\begin{aligned}
\Pr\left[\exists S, T \text{ as above such that } \Gamma(S) \subseteq T\right] &\leq \sum_{S,T} \left(\frac{5ds}{8m}\right)^{ds} \\
&\leq \sum_{s=1}^{n/10d} \binom{n}{s} \binom{m}{5ds/8} \left(\frac{5ds}{8m}\right)^{ds} \\
&\leq \sum_{s=1}^{n/10d} \left(\frac{ne}{s}\right)^{s} \left(\frac{8me}{5ds}\right)^{5ds/8} \left(\frac{5ds}{8m}\right)^{ds} && \left(\binom{n}{k} \geq (ne/k)^k\right) \\
&= \sum_{s=1}^{n/10d} \left(\frac{ne}{s}\right)^{s} e^{5ds/8} \left(\frac{5ds}{8m}\right)^{3ds/8} \\
&\leq \sum_{s=1}^{n/10d} \left(\frac{ne}{s}\right)^{s} e^{5ds/8} \left(\frac{5ds}{6n}\right)^{3ds/8} && (m \geq 3n/4) \\
&= \sum_{s=1}^{n/10d} \left(\frac{s}{n}\right)^{s(3d/8-1)} e^{s(5d/8+1)} \left(5d/6\right)^{3ds/8} \\
&\leq \sum_{s=1}^{n/10d} (10d)^{-s(3d/8-1)} e^{s(5d/8+1)} \left(5d/6\right)^{3ds/8} && (s/n \leq 1/10d) \\
&\leq \sum_{s=1}^{\infty} (10d)^{-s(3d/8-1)} e^{s(5d/8+1)} \left(5d/6\right)^{3ds/8} \\
&= \frac{\alpha}{1-\alpha},
\end{aligned}$$

where $\alpha = (10d)^{1-(3d/8)} e^{(5d/8)+1} (5d/6)^{3d/8}$. The above is less than 1 when $\alpha < 1/2$. To check for what values of d this is true,

$$\begin{aligned}\log\alpha &= \left(1-\frac{3d}{8}\right)(\log 10+\log d)+1+\frac{5d}{8}+\frac{3d}{8}(\log(5/6)+\log d)\\ &= \log d + d\left(\frac{5}{8}-\frac{3}{8}\log(10)+\frac{3}{8}\log(5/6)\right)+(1+\log 10)\\ \frac{\mathrm{d}\log\alpha}{\mathrm{d}\,d} &\approx \frac{1}{d}-0.306,\end{aligned}$$

which is negative for $1/d < 0.306$ (or equivalently, $d \geq 5$). Since α is decreasing in d for $d \geq 24$, it suffices to check that $\alpha < 1/2$ when $d = 24$. Indeed, it is easily verified that $\alpha \approx 0.413 < 1/2$ in this case, completing the proof. □

Now, let us look at reduction of randomness using magical graphs.
Let $\mathcal{A}$ be an algorithm that uses k random bits with error probability $< 1/16$. Take $n = 2^k$, and let $G = (L \sqcup R, E)$ be a (n, n, d)-magical graph. Choose a random vertex $v \in L$, and take all d neighbours $u_1, \ldots, u_d$ of v. Each u_i can be thought of as a k bit string. For $i = 1, \ldots, d$, run $\mathcal{A}$ with u_i as the choice of random bits. Observe that we are only using k random bits here, namely in the choice of v.

Why does the error probability go down?
Let $B \subseteq \{0,1\}^k$ be the set of "bad" inputs for algorithm $\mathcal{A}$. We know that $|B| \leq n/16$. What is the probability of failure when we run it d times as described above? The algorithm fails iff every u_i is in B.
We claim that there are less than $n/10d$ such vertices v with every neighbour in B. Suppose instead that there is a set S with $n/10d$ vertices with all neighbours in B. Then,

$$\frac{n}{16} > |B| \geq \Gamma(S) \geq \frac{5d}{8}|S| \geq \frac{n}{16},$$

which is a contradiction.
Therefore, the probability of failure is at most $1/10d$.
We can make d very large (up to a limit forced by $n = 2^k$), so the probability of failure can be made very small. Using the same number of random bits, we have managed to significantly decrease the error probability.
The issue now however is that the above scheme requires the construction of exponentially large magical graphs. We require a very efficient algorithm (polynomial in $\log n$) to sample the neighbours of a random vertex in a magical graph.