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

Lecture 9: 29-08-2022

Scribe: Aniruddha Joshi *Lecturer: Rohit Gurjar*

1 Magical Graphs and Codes

In the last lecture we saw the definition and existence of **magical graphs**. We will show how good error correcting codes can be constructed using such magical graphs.

Definition 9.1. *A bipartite graph (L, R, E) is a (n, m, d)-magical graph if the following holds*

- $|L| = n$
- $|R| = m$
- $m \geq \frac{3n}{4}$
- $\forall v \in L \ deg(v) = d$
- $\forall S \subseteq L, |S| \leq \frac{n}{10d} \implies |\Gamma(S)| \geq \frac{5d|S|}{8}$

Claim 9.2. *For any $S \subseteq L$ such that $|S| \leq \frac{n}{10d}$, there exists a vertex $v \in \Gamma(S)$ such that v has exactly one neighbour in S.*

Proof. We prove this by contradiction. Since $\Gamma(S)$ is the set of neighbours of S, every vertex $v \in \Gamma(S)$ has at least one neigbour in S. Suppose every vertex v has at least two neighbours in S, then $E(\Gamma(S), S) \geq 2|\Gamma(S)|$ and so $E(\Gamma(S), S) \geq \frac{10d|S|}{8}$, which is a contradiction as $E(\Gamma(S), S) = d|S|$ (because every vertex in S has degree d). □

Now we will look at a construction of an error correcting code.

Definition 9.3 (Linear Codes)**.** *A code $\mathscr{C} \subseteq \{0,1\}^n$ can be viewed as $\mathscr{C} \subseteq \mathbb{F}_2^n$. $\mathscr{C}$ is called a linear code if $\forall x, y \in \mathbb{F}_2^n, (x \in \mathscr{C} \wedge y \in \mathscr{C}) \implies x + y \in \mathscr{C}$. Note that the $+$ operation is in the field $\mathbb{F}_2$.*

Let's consider a (n, m, d)-magical graph, such that $m = \frac{3n}{4}$ (in the last class we saw that such a graph exists).

Definition 9.4. *We define a matrix $M \in \mathbb{R}^{m \times n}$ as follows:*

$$M_{ij} = 1 \text{ iff } i \in R \text{ and } j \in L \text{ are adjacent.}$$

Definition 9.5. *Let us define a code as $\mathscr{C} = \{x : Mx = 0\}$. (Note that the matrix multiplication is over $\mathbb{F}_2$) The matrix M in the above definition is called a parity check matrix.*

Example 1. Let $M = [1, 1, 1]$, the code $\mathscr{C}$ such that $\mathscr{C} = \{x : Mx = 0\}$ is then given as $\mathscr{C} = \{(1,1,0), (1,0,1), (0,1,1), (0,0,0)\}$. Here $d_H(\mathscr{C}) = 2$ and $r(\mathscr{C}) = \frac{2}{3}$.

From definition 9.5, we can see that $|\mathscr{C}| = 2^{n-rank(M)}$ (because $n - rank(M)$ is the dimension of null space and power of 2 because it is in field $\mathbb{F}_2$). So, $r(\mathscr{C}) = \frac{n-rank(M)}{n} \geq \frac{1}{4}$

Claim 9.6. $d_H(\mathscr{C}) > \frac{n}{10d}$

Proof. Since $d_H(x, y) = \alpha \implies d_H(x - y, 0) = \alpha$. Hence, $d_H(\mathscr{C})$ is nothing but the minimum weight (number of nonzero entries) in any codeword other than 0^n. For the sake of contradiction, suppose Z is a (nonzero) codeword with $\frac{n}{10d}$ non-zero entries, then let's define $S = \{v \in L : Z_v = 1\}$, that is S is the set of all columns of M such that their indices have non-zeros entries in Z. Notice that MZ is just the sum of all the column vectors corresponding to S. So, by claim 9.2 there exists a row i such that there is exactly one vector in S which has entry 1 in row i, so $MZ \neq 0$. That means, Z is not a codeword, which is a contradiction. □

2 Undirected Connectivity

Problem 1. Given a graph G, and two vertices $s, t \in G$, is there a path between s and t? (we are looking for a log-space algorithm)

Let us look at an algorithm.
IsPath$(\mathbf{s}, \mathbf{t}, \mathbf{2^\ell})$ outputs whether there is a path between s and t with length at most 2^l.
Algorithm

```
IsPath(s, t, 2^ℓ){
if (s, t) ∈ E then
    return YES;
else
    for v ∈ V do
        if IsPath(s, v, 2^(ℓ−1)) == YES and IsPath(v, t, 2^(ℓ−1)) == YES then
            return YES;
        end
    end
    return NO;
end
}
```

Algorithm 1: Function IsPath

We can see that algorithm 1 works in $O(\log n \times \log n)$ space. We can see this as follows, the maximum depth of the algorithm is $O(\ell) = O(\log n)$ and in each instance, we need $O(\log n)$ space to store the number corresponding to v, s, t and ℓ. Also, note that the algorithm is not polynomial time.

Now we will see a randomized algorithm for the same problem with $O(\log n)$ space.
Algorithm

1. $v \leftarrow s$
2. $u \leftarrow$ a neighbour of v chosen uniformly randomly
3. if $u == t$ return YES
4. otherwise $v \leftarrow u$, go to 1.

Claim 9.7. $\Pr\left[t \text{ is seen within } O(n^3 \log n) \text{ steps}\right] \geq \frac{1}{2}$

We will see the proof of this claim in the next class.

Side note: This algorithm will not work for directed graphs (i.e in $O(n^3 logn)$ steps. There may be an exponentially small probability to reach t from s. Counter-example: consider a family of graphs G_n such that the vertices are $V = \{1, 2, \cdots, n\} \cup \{s, t\}$ and the edges are $E = \{(i, i+1) | i < n\} \cup \{(i, s) | \forall i\} \cup \{(s, 1), (n, t)\}$.

We will only consider undirected graphs for this algorithm.

For simplicity let us assume that the given graph is d-regular (if the graph is not d-regular then add self loops).

Definition 9.8. *Let us define a random walk matrix M such that*

$$M_{ij} = \frac{\delta_{ij}}{d}, \text{ where } \delta_{ij} \text{ is the number of edges between vertex } i \text{ and vertex } j$$

We define a probability vector $v^{(t)} \in \mathbb{R}^n$ such that $v_i^{(t)}$ is the probability of reaching at vertex i at time t. Clearly, for all i, $0 \leq v^{(t)}{}_i \leq 1$ and $\sum_i v_i^{(t)} = 1$. Observe that

$$v^{(t+1)} = Mv^{(t)}.$$

If our walk starts at s then $v_s^{(0)} = 1$ and for $i \neq s$, $v_i^{(0)} = 0$.

So it is easy to see that $v^{(t)} = M^t v^{(0)}$ (does $v^{(t)}$ converge to a probability distribution?)

Observation 1. If $u = (\frac{1}{n}, \frac{1}{n}, \cdots, \frac{1}{n})$, then $Mu = u$

Are there any other stationary distributions?

If G is not connected then there can be several stationary distributions.

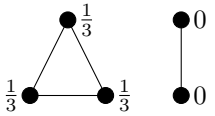

Figure 1: Graph with non-uniform stationary distribution