SOLUTIONS TO ASSIGNMENT 6

Exercise 1 (Random graphs are good expanders)**.** In this exercise, we will show the existence of good expander through a probabilistic method. Recall that a bipartite graph with n left vertices, m right vertices, and left degree D is an $(n, m, D, \gamma, D(1-\varepsilon))$ expander if for all subsets $\mathcal{S}$ of left vertices with $|\mathcal{S}| \leq \gamma n$, we have $|N(\mathcal{S})| > D(1-\varepsilon)|\mathcal{S}|$ where $N(\mathcal{S})$ denotes the set of neighbours of $\mathcal{S}$.

We will prove the following theorem:

Theorem: Fix $0 < \varepsilon < 1$ and $n \geq m$ arbitrarily, let D be a large enough integer to satisfy ($\log$ is to the base 2)

$$D \geq \frac{1}{\varepsilon}\Big(\log\Big(\frac{4e^2}{\varepsilon}\Big) + \log D + \log\Big(\frac{n}{m}\Big)\Big), \tag{1}$$

and let

$$\gamma = \frac{\varepsilon m}{2eDn}.$$

Then, there exist expander graphs with parameters $(n, m, D, \gamma, (1-\varepsilon)D)$.

To prove the theorem, we pick a random bipartite graph $\mathcal{G} = (\mathcal{L}, \mathcal{R}, \mathcal{E})$ as follows. We let $|\mathcal{L}| = n$ and $|\mathcal{R}| = m$ and choose the edges in $\mathcal{E}$ randomly as follows. For every vertex $\ell \in \mathcal{L}$ we pick D random vertices *with replacement* in $\mathcal{R}$ and connect them to ℓ. Note that this implies that we can have multi-edges and so technically the vertices in $\mathcal{L}$ need not be D-regular. We will fix this at the end(*). Let $1 \leq s \leq \lfloor \gamma n \rfloor$ be an integer and let $\mathcal{S} \subseteq \mathcal{L}$ be an arbitrary subset of size exactly s. We will argue that with the chosen parameters, the probability that $|N(\mathcal{S})| < D(1-\varepsilon)s$ is small enough so that even after taking a union bound over all choices of s and $\mathcal{S}$, the probability that all sufficiently small sets expand by a factor of $D(1-\varepsilon)$ is strictly larger than 0. This proves the existence of a graph with the desired properties.

Fix s and $\mathcal{S}$ as above. Let $\mathcal{E}(\mathcal{S}) = \{e_1, e_2, \ldots, e_{sD}\}$ be the sD random choices of edges departing the s vertices in $\mathcal{S}$. It may be helpful for concreteness to choose here a particular labeling order for the e_i's, with say $e_1, e_2, \ldots, e_D$ corresponding to the edges of the top most vertex in $\mathcal{S}$, $e_{D+1}, e_{D+2}, \ldots, e_{2D}$ corresponding to the second vertex in $\mathcal{S}$, and so on. Further, let $\{r_i\}$ denote the set of nodes in $N(\mathcal{S})$. Hence, each vertex in $\mathcal{S}$ is connected through some edge e_i to some vertex $r_{j(i)}$ in $N(\mathcal{S})$. We call an edge e_i (for $i > 1$) a *repeat* if $r_{j(i)} \in \{r_{j(1)}, \ldots, r_{j(i-1)}\}$. Note that if the total number of repeats is at most εsD, then $|N(\mathcal{S})| \geq D(1-\varepsilon)s$. Thus, it suffices to show that the probability of more than εsD repeats is small.

1. Show that the probability that e_i ($i \geq 2$) is a repeat is at most
$$\frac{i-1}{m} \leq \frac{sD}{m}. \tag{2}$$

2. Argue that
$$\Pr[\{e_{a_1}, e_{a_2}, \ldots, e_{a_k}\} \text{ are repeats}] = \prod_{t=1}^{k} \Pr[e_{a_t} \text{ is a repeat} | e_{a_1}, \ldots, e_{a_{t-1}} \text{ are repeats}] \leq \left(\frac{sD}{m}\right)^k \tag{3}$$
(with indices $1 \leq a_1 < a_2 < \ldots < a_k \leq sD$).

3. Justify each step:

$$\begin{aligned}
\Pr\big[\mathcal{E}(\mathcal{S}) \text{ containts at least } \varepsilon sD \text{ repeats}\big] &\leq \Pr\big[\mathcal{E}(\mathcal{S}) \text{ contains a subset of } \varepsilon sD \text{ repeats}\big] \\
&\leq \binom{Ds}{\varepsilon sD}\left(\frac{sD}{m}\right)^{\varepsilon sD} && (4)\\
&\leq \left(\frac{e}{\varepsilon}\right)^{\varepsilon sD}\left(\frac{sD}{m}\right)^{\varepsilon sD} && (5)\\
&= \left(\frac{esD}{\varepsilon m}\right)^{\varepsilon sD} && (6)\\
&= \left(\frac{s}{2\gamma n}\right)^{\varepsilon sD}. && (7)
\end{aligned}$$

4. By taking a union bound over all $\binom{n}{s}$ choices for $\mathcal{S}$, show that the probability that there exists some set $\mathcal{S}$ of size s that does not expand by a factor of $D(1-\varepsilon)$ is at most

$$\left(\frac{1}{2}\right)^{s}. \tag{8}$$

5. Conclude that the probability that G is not an $(n, m, D, \gamma, D(1-\varepsilon))$ bipartite expander is strictly less than 1.

6. Recall that the random graph generation does not guarantee D regularity for left verticies since "for every vertex $\ell \in \mathcal{L}$ we pick D random vertices *with replacement* in $\mathcal{R}$ and connect them to ℓ. " Consider now the slight variation in code generation where each left vertex is connected to a random subset of exactly D left vertices—in other words, each left vertex selects uniformly at random a subset of D right vertices as its neighbors. How does the analysis change?

Solution. 1. The probability that e_i ($i \geq 2$) is a repeat is at most

$$\frac{i-1}{m} \;\leq\; \frac{sD}{m}. \tag{9}$$

To see this, note that the probability that e_i is a repeat equals to the number of distinct vertices in $\{r_{j(1)}, r_{j(2)}, \ldots, r_{j(i-1)}\}$ divided by m, which is at most $(i-1)/m$. Further, since $i \leq sD$, we obtain the above bound, which is uniform in i.

2. Through a similar argument we have

$$\Pr[\{e_{a_1}, e_{a_2}, \ldots, e_{a_k}\} \text{ are repeats}] = \prod_{t=1}^{k} \Pr[e_{a_t} \text{ is a repeat} | e_{a_1}, \ldots, e_{a_{t-1}} \text{ are repeats}] \leq \left(\frac{sD}{m}\right)^{k} \tag{10}$$

(with indices $1 \leq a_1 < a_2 < \ldots < a_k \leq sD$).

3. Hence,

$$\begin{aligned}\Pr\big[\mathcal{E}(\mathcal{S}) \text{ containts at least } \varepsilon sD \text{ repeats}\big] &\le \Pr\big[\mathcal{E}(\mathcal{S}) \text{ contains a subset of } \varepsilon sD \text{ repeats}\big] \\ &\le \binom{Ds}{\varepsilon sD}\left(\frac{sD}{m}\right)^{\varepsilon sD} && (11)\\ &\le \left(\frac{e}{\varepsilon}\right)^{\varepsilon sD}\left(\frac{sD}{m}\right)^{\varepsilon sD} && (12)\\ &= \left(\frac{esD}{\varepsilon m}\right)^{\varepsilon sD} && (13)\\ &= \left(\frac{s}{2\gamma n}\right)^{\varepsilon sD} && (14)\end{aligned}$$

where (11) follows by a union bound over all possible locations of εjD repeats and by (10); where (12) follows from the standard bound on binomial coefficients ($\binom{a}{b} \le \left(\frac{ae}{b}\right)^b$); and where (14) follows from the choice of γ (indeed, $2e\gamma n = \varepsilon m/D$).

4. Taking a union bound over all $\binom{n}{s}$ choices for $\mathcal{S}$, the probability that there exists some set $\mathcal{S}$ of size s that does not expand by a factor of $c(1-\varepsilon)$ is at most

$$\begin{aligned}\binom{n}{s}\left(\frac{s}{2\gamma n}\right)^{\varepsilon jD} &\le \left(\frac{en}{s}\right)^{s}\left(\frac{s}{2\gamma n}\right)^{\varepsilon sD} && (15)\\ &\le \left(\frac{1}{2}\right)^{s} && (16)\end{aligned}$$

It remains to justify (16). This inequality is equivalent to showing that for every $1 \le s \le \gamma n$,

$$\left(\frac{en}{s}\right)\left(\frac{s}{2\gamma n}\right)^{\varepsilon D} \le \frac{1}{2}. \tag{17}$$

Since $\varepsilon D > 1$ by our choice of D (see (1)), the left-hand side is increasing in s, and thus it suffices to check the case $s = \gamma n$. This holds provided that ($\log$ is to the base 2)

$$D \;\ge\; \frac{1}{\varepsilon}\log\left(\frac{2e}{\gamma}\right) = \frac{1}{\varepsilon}\log\left(\frac{4e^2 Dn}{\varepsilon m}\right) = \frac{1}{\varepsilon}\Big(\log\Big(\frac{4e^2}{\varepsilon}\Big) + \log D + \log\Big(\frac{n}{m}\Big)\Big) \tag{18}$$

which is condition (1).

5. Taking a union bound over all $s \le \gamma n$, we conclude that the probability that $\mathcal{G}$ is not an $(n, m, D, \gamma, D(1-\varepsilon))$ bipartite expander is strictly less than 1.

6. To complete the proof, it remains to address (*), namely we need to show that it is possible to generate an expander graph with the same parameters, which is also D-regular. Recall that the random graph generation is such that "for every vertex $\ell \in \mathcal{L}$ we pick D random vertices *with replacement* in $\mathcal{R}$ and connect them to ℓ. " Consider now the slight variation where each left vertex selects uniformly at random a subset of D right vertices as its neighbors. In doing so, (9) and (10) hold, and the rest of the proof remains the same.

Exercise 2 (Minimum distance)**.** Let $\mathcal{G}$ be an $(n, m, D, \gamma, D(1-\epsilon))$ be an expander graph for some $0 < \epsilon < 1/2$. Given any set of left vertices $\mathcal{S}$, a right vertex v is said to be a unique neighbour of $\mathcal{S}$ if it is adjacent to exactly one vertex in $\mathcal{S}$. Let $U(\mathcal{S})$ denote the set of unique neighbours of $\mathcal{S}$.

1. Fix any set of left vertices $\mathcal{S}$ such that $|\mathcal{S}| \leq \gamma n$. How many edges leave $\mathcal{S}$? Using this, compute an upper bound on the number of vertices in $N(\mathcal{S})$ that have more than one incident edge from $\mathcal{S}$.

2. Use the above to argue that $|U(\mathcal{S})| \geq D(1-2\epsilon)|\mathcal{S}|$.

3. Use the second part to argue that the minimum distance of the corresponding expander code is at least γn.

   Hint: Choose any nonzero codeword and label the left vertices by the codeword bits. Let $\mathcal{S}$ be the support set of vertices labelled 1. What can you say about $U(\mathcal{S})$?

4. Using similar arguments (in particular by showing that $|U(\mathcal{S})| > 0$), conclude that the minimum distance is at least $2\gamma(1-\epsilon)n$.

   Hint: Assume that there exists $T \subset \mathcal{S}$ with $|T| = \gamma n$. Show that

   $$|U(\mathcal{S})| \geq |U(T) - N(\mathcal{S} \backslash T)| > 0.$$

Solution. 1. The number of edges leaving $\mathcal{S}$ is $D|\mathcal{S}|$. Since the graph is an expander, $\mathcal{S}$ has at least $(1-\epsilon)D|\mathcal{S}|$ neighbours. Since there are $D|\mathcal{S}|$ edges, by the pigeonhole principle, at most $\epsilon D|\mathcal{S}|$ neighbours of $\mathcal{S}$ can have more than one incident edge from $\mathcal{S}$.

2. $\mathcal{S}$ has at least $(1-\epsilon)D|\mathcal{S}|$ neighbours, of which at most $\epsilon D|\mathcal{S}|$ can have more than one incident edge from $\mathcal{S}$. Therefore, $|U(\mathcal{S})| \geq (1-2\epsilon)D|\mathcal{S}| > 0$ since $\varepsilon < 1/2$.

3. Suppose by contradiction, that the minimum distance is $\leq \gamma n$. Since this is a linear code, this means that the minimum codeword weight is $\leq \gamma n$. Now pick a codword with minimum weight and let $\mathcal{S}$ be the its support, that is the set of vertices labelled 1. Since this is a valid codeword, $U(\mathcal{S})$ should be empty—this is because any check node in $U(\mathcal{S})$, has only one neighbor in $\mathcal{S}$, hence the equation of this check node cannot be satisfied if $U(\mathcal{S})$ is non-empty. However, 2. tells us that $|U(\mathcal{S})| > 0$. By contradiction this implies that $|\mathcal{S}|$ is too small to be the support of a valid codeword. Hence, the minimum distance is at least γn.

4. See Theorem 11.3.4 in the textbook.

Exercise 3 (Encoding/decoding complexity of expander codes)**.** Expander codes have low encoding and decoding complexity.

- What is the encoding complexity of an expander code?
- What is the computational complexity in each iteration of decoding an expander code? Justify first that it can be made $O(n^2)$, then improve your method to make it $O(n)$.

Solution. 1. An expander code is linear. Hence, the encoding complexity is $O(n^2)$.

2. In each iteration, we need to find a left vertex which has more unsatisfied neighbours than satisfied ones, and also update the parities at the right vertices. The complexity is $O(n)$.

   At each iteration the bit-flipping algorithm takes $O(n)$ time to find a variable with a number of violated clauses larger than the number of correct clauses. At each step the total number of violated clauses decreases by at least one. Hence, the total number of steps is $O(m)$ and hence the overall decoding complexity is $O(nm)$.

   Let us improve this method by improving upon the search at each iteration. For this argument we are going to assume that the maximum right degree is bounded by a constant r.

   - Preprocessing step: compute the value at each check $\rightarrow$ $O(m \cdot r)$, compute at each variable the number of satisfied/unsatisfied clauses and produce the list $\mathcal{L}$ of variables with more unsatisfied clauses than satisfied clauses $\rightarrow O(n \cdot D)$.
   - At each iteration, we select a variable from $\mathcal{L}$, and flip its value. We update the list of unsatisfied checks in $O(D)$ time, and update $\mathcal{L}$ in $O(Dr)$ (add or remove new checks).

   Hence, each step takes $O(Dr)$ time, and since the number of iterations is at most m this implementation of the algorithm takes $O(mDr) = O(m)$ whenever D and r are constant.