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 $(n, m, D, \gamma, \alpha)$ expander if for all subsets S of left vertices with $|S| \leq \gamma n$, we have $|N(S)| > \alpha|S|$ where $N(S)$ denotes the set of neighbours of S.

We will prove the following: Fix $0 < \varepsilon < 1$, $n \geq m$, $q > 2$, and let D be (implicitly) defined as the solution of

$$D = \frac{\log_{1/(1-\varepsilon)}\left(\frac{qe^{1/\varepsilon+1}Dn}{m}\right)}{\varepsilon}.$$

Let $\alpha = (1-\varepsilon)D$, and let $\gamma = \frac{m}{nDe^{1/\varepsilon}}$. Then, there exist expander graphs with parameters $(n, m, D, \gamma, (1-\varepsilon)D)$.

Pick a random graph in the following manner: For each left vertex, pick D neighbours uniformly at random from the set of all $\binom{m}{D}$ subsets of right vertices. This is done independently for each vertex. Call the resulting random graph $\mathcal{G}$. We want to show that if the parameters are chosen as above then

$$\Pr[\mathcal{G} \text{ is not an } (n, m, D, \gamma, \alpha) \text{ expander}] < 1.$$

1. Choose any set of left vertices S and set of right vertices T, with $|S| = s \leq \gamma n$ and $|T| = t \leq \alpha s$. Compute the probability that $N(S) \subset T$.

2. Argue that

   $$\Pr[\mathcal{G} \text{ is not an } (n, m, D, \gamma, \alpha) \text{ expander}] = \Pr[\exists S \subset \mathcal{L}, T \subset \mathcal{R} : |S| \leq \gamma n, |T| \leq \alpha|S|,\ N(S) \subset T]$$

   where $\mathcal{L}, \mathcal{R}$ denote the set of left and right vertices respectively.

3. Use the first two parts to get an upper bound on the probability that $\mathcal{G}$ is not an expander.

4. Using the bound $\binom{a}{b} \leq \left(\frac{ea}{b}\right)^b$, prove that as long as $m > 3n/4$, $D > 32$, $\gamma = \frac{3}{32}$, $\alpha = 5D/8$, the probability that $\mathcal{G}$ is not an expander is < 1.

Solution. 1. Any left vertex y chooses D distinct neighbours $A_1, A_2, \ldots, A_D$ uniformly at random, and independently of the other vertices. Now define $B_1, B_2, \ldots, B_D$ similarly as $A_1, \ldots, A_D$ except that the $B_i's$ are independent of each other.

We have

$$\begin{aligned}
\Pr[N(y) \subset T] &= \Pr[A_1(y), A_2(y), \ldots, A_D(y) \subset T] \\
&= \Pr[\{B_1 \in T\} \cap \ldots \cap \{B_D \in T\} \cap \{B_1, \ldots B_D \text{ are all distinct}\}] \\
&\leq \Pr[\{B_1 \in T\} \cap \ldots \cap \{B_D \in T\}] \\
&= (\Pr[\{B_1 \in T\}])^D \\
&= (t/m)^D
\end{aligned}$$

Hence, each vertex in S has probability at most $(t/m)^D$ to have all its neighbors in T. Since each vertex in S choses its neighbors independently we get

$$\Pr[N(S) \subset T] \leq \left(\frac{t}{m}\right)^{sD}$$

2. The random graph $\mathcal{G}$ is not an expander if and only if there exists some subset S of left vertices with $|S| \leq \gamma n$ whose neighbourhood is of size less than or equal to $\alpha|S|$.

3. We have

$$\begin{aligned}
\Pr[\mathcal{G} \text{ is not a } (\gamma,\alpha) \text{ expander}] &= \Pr[\exists S \subset \mathcal{L}, |S| \leq \gamma n, \exists T \subset \mathcal{R}, |T| \leq \alpha|S| : N(S) \subset T] \\
&\leq \Pr[\exists S \subset \mathcal{L}, |S| \leq \gamma n, \exists T \subset \mathcal{R}, |T| = \alpha|S| : N(S) \subset T] \\
&\leq \sum_{S \subset \mathcal{L}, |S| \leq \gamma n} \sum_{T \subset \mathcal{R}, |T| = \alpha|S|} \Pr[N(S) \subset T] \\
&\leq \sum_{s=1}^{\gamma n} \binom{n}{s} \binom{m}{\alpha s} \left(\frac{\alpha s}{m}\right)^{sD}
\end{aligned}$$

4. We have

$$\begin{aligned}
\sum_{s=1}^{\gamma n} \binom{n}{s} \binom{m}{\alpha s} \left(\frac{\alpha s}{m}\right)^{sD} &= \sum_{s=1}^{\gamma n} \left(\frac{ne}{s}\right)^{s} \left(\frac{me}{\alpha s}\right)^{\alpha s} \left(\frac{\alpha s}{m}\right)^{sD} \\
&= \sum_{s=1}^{\gamma n} x_s^s \\
&\leq \sum_{s=1}^{\infty} x^s \\
&= \frac{x}{1-x} \\
&< 1
\end{aligned}$$

if for all $s \leq \gamma n$,

$$x_s \triangleq \left(\frac{ne}{s}\right) \left(\frac{me}{\alpha s}\right)^{\alpha} \left(\frac{\alpha s}{m}\right)^{D} \leq x < 1/2.$$

We have

$$\begin{aligned}
x_s &= \left(\frac{ne}{s}\right) \left(\frac{me}{\alpha s}\right)^{\alpha} \left(\frac{\alpha s}{m}\right)^{D} \\
&\leq \frac{ne}{s} \left(\frac{(1-\varepsilon) \cdot D \cdot s \cdot e^{1/\varepsilon}}{m}\right)^{\varepsilon D} \qquad \text{since} \quad \alpha = (1-\varepsilon)D, \quad e^{(1-\varepsilon)D} \leq e^{D} \\
&\leq \frac{e}{\gamma} \left(\frac{(1-\varepsilon) \cdot D \cdot \gamma \cdot n \cdot e^{1/\varepsilon}}{m}\right)^{\varepsilon D} \qquad \text{since} \quad s \leq \gamma n \\
&= \left(\frac{n \cdot D \cdot e^{1+1/\varepsilon}}{m}\right) (1-\varepsilon)^{\varepsilon D} \qquad \text{by setting} \quad \gamma = \frac{m}{n \cdot D \cdot e^{1/\varepsilon}}
\end{aligned}$$

Setting

$$D = \frac{\log_{1/(1-\varepsilon)}\left(\frac{qe^{1/\varepsilon+1}Dn}{m}\right)}{\varepsilon}$$

with $q > 2$ we have $x \leq 1/q < 1/2$ and expanders with such parameters exist. Note here that D is implicitly defined and is related to the Lambert W function, which is the reciprocal of the function we^w.

Reference: "Expander graphs and their applications" by S. Hoory, N. Linial, A. Wigderson, `https://www.cs.huji.ac.il/~nati/PAPERS/expander_survey.pdf`

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 S, a right vertex v is said to be a unique neighbour of S if it is adjacent to exactly one vertex in S. Let $U(S)$ denote the set of unique neighbours of S.

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

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

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 S be the support set of vertices labelled 1. What can you say about $U(S)$?

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

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

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

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

2. S has at least $(1-\epsilon)D|S|$ neighbours, of which at most $\epsilon D|S|$ can have more than one incident edge from S. Therefore, $|U(S)| \geq (1-2\epsilon)D|S| > 0$ 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 S be the its support, that is the set of vertices labelled 1. Since this is a valid codeword, $U(S)$ should be empty—this is because any check node in in $U(S)$, has only one neighbor in S, hence the equation of this check node cannot be satisfied if $U(S)$ is non-empty. However, 2. tells us that $|U(S)| > 0$. By contradiction this implies that $|S|$ 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.