Complexity and Advanced Algorithms – Assignment 6

Siddharth Bhat (20161105)

November 18, 2018

1 Existence of expander graph

1.1 problem

- $|L| = |R| = n$
- Every vertex in L has degree $n^{\frac{3}{4}}$, every vertex in R has degree at most $3n^{\frac{3}{4}}$
- Ever subset of $n^{\frac{3}{4}}$ vertices in L has at least $n - n^{\frac{3}{4}}$ vertices in R.

1.2 Solution

Let each vertex in L pick $n^{\frac{3}{4}}$ neighbors uniformly at random. We merge repeat picks of a neighbor from R into a single neighbor.

Probability of having exactly $n^{\frac{3}{4}}$ neighbors is:

$$P_1 = \frac{{}^{n}C_{n^{\frac{3}{4}}}}{n^{n^{\frac{3}{4}}}}$$

2 Concave functions problem

2.1 Problem

Let f be a concave function and g be a linear function such that $g(0) \leq f(0)$ and $g(1) \leq f(1)$. Show that in $[0, 1]$, $g(x) \leq f(x)$.

2.2 Solution

Recall that f being concave implies that:

$$f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y)$$

Therefore, let $\lambda \in [0,1]$. Now, consider an aribitrary point in $[0,1]$ as

$$\lambda = (1-\lambda)\cdot 0 + \lambda \cdot 1$$

. Evaluate f at this point and show that it upper bounds g.

$$\begin{aligned}
&f(\lambda) = f((1-\lambda)\cdot 0 + \lambda \cdot 1) \\
&\geq \lambda f(1) + (1-\lambda) f(0) \\
&\big(\text{Since } f(1) \geq g(1), f(0) \geq g(0)\ \big), \\
&\geq \lambda g(1) + (1-\lambda) g(0) \\
&\geq g(\lambda \cdot 1 + (1-\lambda)\cdot 0) = g(\lambda)
\end{aligned}$$

Hence, $f(\lambda) \geq g(\lambda),\ 0 \leq \lambda \leq 1$

3 Problem 11.1 - Sampling a circle

3.1 Problem

Consider a circle of diameter 1, enclosed within a square of side length 1. Sample points uniformly and independently from the square. Set

$$X_t \equiv 1 \text{ if } t \text{ th point is inside circle, } 0 \text{ otherwise}$$

.

$$\mathbb{P}\Big[X_t = 1\Big] = \pi\left(\frac{1}{2}\right)^2 = \frac{\pi}{4}$$

We now define

$$\begin{aligned}
X &= \sum_{i=1}^{N} X_i \\
\mu_X &= \mathbb{E}\big[X\big] = \frac{N\pi}{4}
\end{aligned}$$

Give an upper bound on N for which $\frac{4X}{N}$ gives an estimate of π that is accurate upto d digits, with probability at least $1-\delta$.

3.2 Solution

To be accurate upto d digits, two values x and y must have a distance of at most 10^{-d}. For example, to be accurate upto 1 digit, they must be off by at most $0.0\overline{9} = 0.1$.

Notice the expectation of Pi is:

$$\mu_{Pi} = \mathbb{E}\big[Pi\big] = \frac{4\mathbb{E}\big[X\big]}{N} = \frac{4}{N}\cdot\frac{N\pi}{4} = \pi$$

We want to make sure that our estimate of Pi is within 10^{-d} of π. That is,

$$
\begin{aligned}
&|Pi - \pi| \leq 10^{-d} \\
&-10^{-d} \leq Pi - \pi \leq 10^{-d} \\
&\pi - 10^{-d} \leq Pi - \pi \leq \pi + 10^{-d} \\
&\pi(1 - \frac{10^{-d}}{\pi}) \leq Pi - \pi \leq \pi(1 + \frac{10^{-d}}{\pi}) \\
&\mu_X(1 - D) \leq X \leq \mu_X(1 + D) \text{ where } D = \frac{10^{-d}}{\pi}
\end{aligned}
$$

This allows us to apply Chernoff bounds as follows:

$$\mathbb{P}\Big[X \leq \mu_X(1 + D)\Big] = 1 - \mathbb{P}\Big[X \geq \mu_X(1 + D)\Big] = 1 - e^{\frac{-\mu d^2}{4}}$$

We have $\delta = e^{\frac{-\mu D^2}{4}}$. We need to compute bounds on N, as follows:

$$
\begin{aligned}
&\delta = e^{\frac{-\mu D^2}{4}} \\
&\log\delta = \frac{-\mu D^2}{4} \\
&\frac{-4\log\delta}{D^2} = \mu \\
&\frac{-4\log\delta}{D^2} = \frac{N\pi}{4} \\
&\frac{-16\log\delta}{D^2\pi} = N \\
N = &\frac{-16\log\delta}{(\frac{10^{-d}}{\pi})^2\pi}
\end{aligned}
$$

Hence, we have N in terms of d.

4 DNF Counting

4.1 Formula with m clauses and n variables that uniform sampling need exponential time to get a good estimate

.

Let us have n literals $x_1, x_2 \dots x_n$. Denote the clause

$$C_{n,i_0,i_1} \equiv x_1 \wedge \dots x_{i_0-1} \wedge x_{i_0+1} \wedge \dots \wedge x_{i_1-1} \wedge x_{i_1+1} \wedge \dots x_n$$

That is, $C_{n,i_0,i_i,\dots i_d}$ is the clause missing the variable x_{i_0}, x_{i_1}, $x_{i_d}$from variables $x_1 \dots x_n$.

For the clause $C_{n,i_0,i_1,\dots i_d}$ to be satisifed, we need $x_j = 1$ where $j \neq i_0, i_1, \dots i_d$, and $x_{i_0}, x_{i_1}, \dots x_{i_d} \in 0, 1$. So, a C_i has 2^d possible truth assignments.

Now, consider the boolean formula

$$F_n \equiv C_{n,1}$$

Now, the formula has a 2 truth assignments, while the total number of assignments are 2^n. Hence, this is a formula where we will need an exponential number of samples to be able to get a good estimate on the count of the number of solutions.

4.2 How does importance sampling become effective in the above example

$$F_n \equiv C_{n,1,2} \vee C_{n,1,3}$$

Notice that we have an overlap of possible truth assignments for the first and second clause, so we cannot simply add the number of satisfying truth assignments for each clause. For example, the truth assignment $x_i = 1$ satisfies both clauses, as does the truth assignment $x_1 = 0, x_i = 1, i > 0$. In order to accurately count the number of truth assignments, we would have to rely on inlusion exclusion.

When we use importance sampling, our sample space is now the multiset of all possible satisfying assignments for all clauses. It's a multiset because the same satisfying assignment can work for multiple clauses (as exhibited above), and it is important to count such assignments multiple times to make sure the probabilities work out. We will consider a sample of an assignment as a success only if it satisfies the *earliest/smallest* clause.

Due to this, our sample space is drastically cut down: We are only sampling from the collection of all *satisfying truth assignments* per clause, which is polynomial in this case.

4.3 Write an explicit formula with m clauses and n variables such that uniform sampling requires polynomial time (or less)

Consider the formula

$$F_n \equiv C_{n,1,2} \vee C_{n,1,3} \vee C_{n,1,4} \cdots \vee C_{n,1,m}$$

Here, the number of satisfying assignments for each $C_{n,1,i}$ is 4. The total number of satisfuing assignments is upper bounded by the sum of the possible satisfying assignments for each clause, which is $4n$. Since this is our sample space, we will take $poly(n)$ time to uniformly sample from this.