Combinatorics of Computation Spring 2025

Some Randomized Algorithms

Prof. Jara Uitto *Antonio Cruciani*

We will show how to use the tools discussed in the first lesson to analyze some randomized algorithms.

1 Again on random graphs

Let $G \sim G(n,p)$ be the Erdős-Rényi random graph with n vertices and edge appearance probability $p \in [0,1]$. Let T be the number of triangles (i.e., sets of 3 vertices with all 3 edges between them present) in G. What is the expected number of triangles in $G(n,p)$ as a function of n and p? For each triple $\{i,j,k\} \subseteq [1,n]$, define the indicator random variable:

$$X_{i,j,k} = \begin{cases} 1 & \text{If all 3 edges } (i,j),(j,k),(k,i) \in E \\ 0 & \text{Otherwise} \end{cases}$$

Then the number of triangles is the summation over all the possible triples

$$T = \sum_{\{i,j,k\} \in \binom{[1,n]}{3}} X_{i,j,k}$$

Where $\binom{[1,n]}{3} = \{\{i,j,k\} \subseteq [1,n] : i < j < k\}$. Since each $X_{i,j,k}$ is a Bernoulli random variable with

$$\mathbf{E}[X_{i,j,k}] = \Pr\left((i,j),(j,k),(k,i) \in E\right) = p^3$$

Then we can compute the expected number of triangles using the linearity of expectation

$$\mathbf{E}[T] = \binom{n}{3} \cdot p^3$$

And what about 4-cliques?

As exercise.

2 An $(7/8)$ approximation for MAX 3SAT

Consider the MAX-3SAT Problem:

Input: A collection of clauses $C_1, C_2, \ldots, C_m$ over n variables in 3-CNF (e.g. $F = (x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee \overline{x_1} \vee x_2)$

Goal: Find a truth assignment to $x_1, \ldots, x_n$, that satisfies the maximum number of clauses.

Idea: Randomly and independently assign T/F values to $x_1, \dots x_n$.

We can show, that this simple technique gives a (7/8) approximation for the MAX-3SAT problem.

Let

$$X_i = \begin{cases} 1 & \text{If clause } i \text{ is T} \\ 0 & \text{Otherwise} \end{cases}$$

The $\Pr(X_i = 0) = \left(\frac{1}{2}\right)^3$ Thus, $\Pr(X_i = 1) = 1 - \left(\frac{1}{2}\right)^3 = \frac{7}{8}$ Let $X = \sum_{i=1}^{m} X_i$, this is the number of clauses that are satisfied

$$\mathbf{E}[X] = \sum_{i=1}^{m} \frac{7}{8} = \frac{7}{8} \cdot m$$

Definition 1. *An algorithm $\mathcal{A}$ for a maximization problem achieves an approximation factor $\alpha \leq 1$ if for all inputs we have:*

$$\frac{OPT(\mathcal{A})}{OPT^{(\star)}} \geq \alpha$$

Where $OPT^{(\star)}$ is the optimum for the given problem and $OPT(\mathcal{A})$ is the solution computed by algorithm $\mathcal{A}$.

3 Single Source Gossip

Assume you are on a clique of n nodes. And that a node s has a message m. s wants to send its message to all its neighbors (i.e., nodes in the graph). At each round t, s is allowed to pick one of its neighbors $v \in N(s)$ uniformly at random and send its message to v. What is the expected time that s needs to inform all its neighbors?

We can model this problem as a coupon collector problem. Indeed, we can define T as the total number of rounds we need to inform all the neighbors. Now, we are interested in finding $\mathbf{E}[X]$. As for coupon collector, let us define X_i be number of rounds to inform $i-1$ different nodes.

At that point, there are $n-i$ informed nodes, and the probability of contacting one of them is

$$p_i = 1 - \frac{i-1}{n} = \frac{n-(i-1)}{n}$$

Thus X_i's expected value is

$$\mathbf{E}[X_i] = \frac{1}{p_i} = \frac{n}{n-i+1}$$

Hence, the total expected time T is

$$\mathbf{E}[T] = \sum_{i=1}^{n-1} \mathbf{E}[X_i] = \sum_{i=1}^{n-1} \frac{n}{n-i+1} = n \cdot \sum_{i=2}^{n} \frac{1}{i} = n \cdot (H_n - 1) = \mathcal{O}(n \log n)$$

4 Triangle Counting

Algorithm 1: Triangle Counting Estimator

Input: Graph $G = (V, E)$, integer k
Output: Estimated triangle count T_v for each vertex $v \in V$

1 **foreach** $v \in V$ **do**
2     $T_v \leftarrow 0$;
3 **for** $i = 1$ *to* k **do**
4   sample edge (u, v) uniformly at random from E;
5   **foreach** $w \in N(u)$ **do**
6   **if** $w \in N(v)$ **then**
            ; `//` (u, v, w) `forms a triangle`
7             $T_v \leftarrow T_v + \frac{m}{k}$;
8 **return** $\frac{1}{3}\sum_{v \in V} T_v$;

Let us now analyze Algorithm 1. For starters, let us notice that the sum of the number of triangles incident to a vertex v, Δ_v, is equal to 3 times the total number of triangles in the graph:

$$\sum_{v \in V} \Delta_v = 3\Delta$$

That is because each triangle contributes exactly 1 to the triangle count of each of its 3 vertices. Formally, let $\mathcal{T}$ denote the set of all triangles in the graph:

$$\mathcal{T} = \{\, \{x, y, z\} \subseteq V \mid (x, y), (y, z), (z, x) \in E \,\}.$$

By definition, $|\mathcal{T}| = \Delta$. We can rewrite:

$$\sum_{v \in V} \Delta_v = \sum_{v \in V} \sum_{t \in \mathcal{T}} \mathbf{1}[v \in t] = \sum_{t \in \mathcal{T}} \sum_{v \in V} \mathbf{1}[v \in t].$$

Observe that for each triangle $t = \{x, y, z\}$, we have:

$$\sum_{v \in V} \mathbf{1}[v \in t] = |\{x, y, z\}| = 3.$$

Thus:

$$\sum_{t \in \mathcal{T}} \sum_{v \in V} \mathbf{1}[v \in t] = \sum_{t \in \mathcal{T}} 3 = 3|\mathcal{T}| = 3\Delta.$$

Let us now focus on the sampling algorithm, for each iteration of the For loop in line 3, the algorithm

- Picks an edge (u, v) form E with probability $1/m$
- For each vertex w in the neighborhood of u checks whether it shares an edge with the node v. If this is the case, then the number of triangles incident in T_v is incremented by this magic value m/k.

Let us analyze the expected value of T_v. Observe that T_v is a random variable that is increased with probability $1/m$.

$$\mathbf{E}[T_v] = \sum_{i=1}^{k} \Pr(\text{Sample } (v,u) \text{ from } E) \sum_{w \in N(v)} \mathbf{1}[(w,u) \in E] \cdot \frac{m}{k} = k \cdot \frac{1}{m} \cdot \Delta_v \cdot \frac{m}{k} = \Delta_v$$

T_v is said to be an unbiased estimator of Δ_v. Now define,

$$\hat{T} = \frac{1}{3} \sum_{v \in V} T_v$$

Let us compute its expectation,

$$\mathbf{E}\left[\hat{T}\right] = \mathbf{E}\left[\frac{1}{3}\sum_{v \in V} T_v\right] = \frac{1}{3}\mathbf{E}\left[\sum_{v \in V} T_v\right] = \frac{1}{3}\sum_{v \in V} \mathbf{E}[T_v] = \frac{1}{3}\sum_{v \in V} \Delta_v = \frac{1}{3}(3\Delta) = \Delta.$$

References

[1] Mitzenmacher, Michael and Upfal, Eli, *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis, Cambridge university press, 2017*,

[2] Kleinberg, Jon and Tardos, Eva, *Algorithm Design,Addison-Wesley Longman Publishing Co., Inc.m 2005*