COMS 4995: Advanced Algorithms Jan 22, 2020

Lecture 1 – Introduction, Schedule, Approximate Counting

Instructor: *Alex Andoni* Scribes: *Yiming Wang, Zhaoyang Wang*

1 Introduction and Schedule

1.1 Topics

In this class, we will frequently talk about randomized algorithms (e.g. P_A(algorithm correct) $\geq 90\%$, P_A(has runtime O(n)) $\geq 90\%$) as well as approximate algorithms (i.e. if α is the correct answer, algorithm A has approximation answer if it outputs $\hat{\alpha}$ s.t. $\alpha \leq \hat{\alpha} \leq a\alpha$, where a is constant). A list of specific topics is shown below.

1. hashing
   - $h : u \to [n]$ ($[n] = \{1, 2, 3, ..., n\}$)
   - perfect hashing $O(1)$ worst case per query
   - consistent hashing
2. graph algorithms
   - combinatorial in nature
   - max-flow
3. spectral graph algorithms, linear algebra techniques
   - A(n x n matrix) = adjacency matrix of G, $A_{ij} = 1$ iff (i, j) is an edge
   - look at spectral decomposition of A
   - random walk
   - spectral partitioning / clustering
4. optimization
   - linear programming
   - duality
   - polynomial time algorithm for linear programming
   - iterative method (gradient descent, second order method)
   - interior point method
   - multiplicative weights update, online algorithm
5. large scale models

- cache models (CPU(register) $\xleftrightarrow{\text{fast}}$ cache $\xleftrightarrow{\text{slow}}$ memory)
- parallel algorithms (map-reduce)

6. extra

   - fast fourier transform (FFT)
   - elliptic-curve cryptography (ECC)
   - exhaustive search

1.2 Prerequisite

1. math

   - probability theory
   - linear algebra

2. CS/algorithm

   - $O()$, $\Omega()$
   - runtime analysis
   - sorting, binary search, basic graph algorithms

1.3 Deliverables

1. scribe one lecture

   - 10%
   - due next day midnight

2. homework

   - 55%
   - 5 in total (1 in every 2 weeks)
   - late policy: 5 late days in total, 10% penalty per day up to 7 days

3. project

   - 35%
   - options
     - reading based / survey
     - implementation based
     - research oriented
   - Progress Report 1 (5%) (a few pages)
   - Progress Report 2 (5%) (a few pages)
   - Final Project (25%) (10 pages)

2 Counting and Morris' Algorithm

Let n be the number of objects or ticks we would like to count. What is the space complexity required to count to n? It turns out that $O(log(n))$ bits is required to exactly count to n.

Theorem 1. *To exactly count to n, we cannot do better than* $\Omega(log(n))$.

However, if the task is approximate counting rather than exact counting, then the problem can be solved much more efficiently via **Morris' algorithm**. The key idea is to use randomized algorithm in order to achieve $O(log(log(n))$ space complexity. The details of the algorithm is shown below:

1. Initialize the counter X to 0
2. For each tick, update $X \leftarrow X + 1$ with probability 2^{-X}, and leave X unchanged with probability $1 - 2^{-X}$
3. output the estimate $\hat{n} = 2^X - 1$

3 Probability

Definition 2 (Expectation). *For a discrete random variable X, the expectation of X is*

$$E[X] = \sum_a aP[X = a]$$

For a continuous random variable X, the expectation of X is

$$E[X] = \int af(a)da$$

where f is the probability density function (PDF) of X.

Lemma 3 (Linearity of Expectation). *Let X and Y be two random variables, then*

$$E[X + Y] = E[X] + E[Y]$$

Lemma 4 (Markov's inequality). *Let X be a non-negative random variable. For all* $\lambda > 0$,

$$P[X > \lambda] \leq \frac{E[X]}{\lambda}$$

Definition 5 (Variance). *Let X be a random variable. The variance of X is*

$$Var[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

Lemma 6 (Chebyshev's Inequality). *Let X be a random variable. For all* $\lambda > 0$,

$$P[|X - E[x]| > \lambda] \leq \frac{Var[X]}{\lambda^2}$$

4 Analysis of Morris' Algorithm

Define X_n as the value of counter X after n ticks, and define $X_0 = 0$. We want the show that the expectation of the output $\hat{n}$ is in fact the actual count.

Claim 7. $E[2^{X_n} - 1] = n$

Proof. We will prove by induction. First, we check the the base cases: for $n = 0$, $E[2^{X_0} - 1] = 0$; for $n = 1$, $E[2^{X_1} - 1] = 1$. Now assume for inductive hypothesis that $E[2^{X_{n-1}} - 1] = n - 1$; we want to show $E[2^{X_n} - 1] = n$. We have:

$$\begin{aligned}
E[2^{X_n} - 1] &= E_{X_n, X_{n-1}, \ldots, X_1}[2^{X_n} - 1] \\
&= E_{X_{n-1}, \ldots, X_1}[E_{X_n}[2^{X_n} - 1]] \\
&= E_{X_{n-1}, \ldots, X_1}[2^{-X_{n-1}}(2^{X_{n-1}+1} - 1) + (1 - 2^{-X_{n-1}})(2^{X_{n-1}} - 1)] \\
&= E_{X_{n-1}, \ldots, X_1}[2 - 2^{-X_{n-1}} + 2^{X_{n-1}} - 1 - 1 + 2^{-X_{n-1}}] \\
&= E_{X_{n-1}, \ldots, X_1}[2^{-X_{n-1}}] \\
&= E[2^{-X_{n-1}} + 1 - 1] = n - 1 + 1 = n
\end{aligned}$$

□