Lecture 2:

Sublinear Algorithms I

Course: Algorithms for Big Data

Instructor: Hossein Jowhari

Department of Computer Science and Statistics
Faculty of Mathematics
K. N. Toosi University of Technology

Spring 2021

Outline

- Sublinear time algorithms: definitions
- A problem in 0,1 Matrices
- The celebrity problem
- Estimating the average degree

Sublinear Time Algorithms

Definitions, Different Types

Definition: A sublinear time algorithm is an algorithm whose running time is sublinear in terms of the input size.

Examples: Input Size $= n$

- $O(n^{0.99})$
- $O(\frac{n}{\log\log n})$
- $O(\log^2 n)$
- $O(\frac{1}{\epsilon^2}\sqrt{n})$ when $\epsilon = \omega(n^{-1/4})$
- $O(\frac{1}{\epsilon^2})$ when $\epsilon = \omega(n^{-1/2})$
- ...

A sublinear time algorithms does not read the whole input!

Sublinear Time Algorithms

Definitions, Different Types

Two types of sublinear time algorithms:

- Algorithms that compute/approximate a target value

  Examples target values: Frequent Items, Average Degree, Statistical Measures, Diameter, Count of Triangles, Cluster Centers, etc

- Property Testers:
  Distinguishing inputs that have a certain property from inputs that are far away from having that property

  Examples: Testing Sortedness, Graph Planarity, Graph Bipartiteness, etc

Warm-up: An 0,1 Matrix Problem

- Suppose we have an m by m $(0,1)$ matrix A.
- Every row of A is sorted. The 0's precede the 1's.
- We want to find a row with most number of 0's.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- Brute-Force solution: read every row. $O(m^2)$ worst-case running time.

Warm-up: An 0,1 Matrix Problem

- Suppose we have an m by m $(0, 1)$ matrix A.
- Every row of A is sorted. The 0's precede the 1's.
- We want to find a row with most number of 0's.

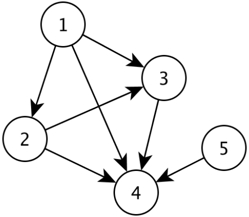
$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- Brute-Force solution: read every row. $O(m^2)$ worst-case running time.
- A sublinear solution: Begin from the first row. Upon seeing a 0 go left, when you see a 1, go down. $O(m)$ running time.

The Celebrity Problem

Definition: Celebrity is a person whom everybody knows but he knows nobody.

Problem: Find a celebrity in a directed graph on n nodes. We are allowed to ask questions like "Does an edge exist from x to y?"

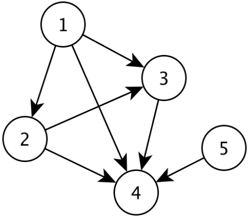


$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The Celebrity Problem

Definition: Celebrity is a person whom everybody knows but he knows nobody.

Problem: Find a celebrity in a directed graph. We are allowed to ask questions like "Does an edge exist from x to y?"



$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

There is a strategy that asks at most $n-1$ questions!

Hint: Every question eliminates a person.

The Celebrity Problem

A question to think about

Definition: A (k,t)-celebrity is a person who whom at least $n - k$ person knows but he knows at most t person.

Problem: Can we find a (k,t)-celebrity in a directed graph on n nodes using sublinear number of edge queries? For what ranges of k and t?

Estimating the average degree in a graph

Input: An undirected connected graph $G = (V, E)$ with n nodes and m edges. (We do not know m !)

Problem: Estimate the average degree $d = \frac{2m}{n}$ using sublinear number of degree queries. Given a vertex $u \in V$, we can query for its degree.

Motivation: A huge social network (people with friends). How many friends each person has in average?

Strategy: We sample a random subset of vertices $S \subseteq V$ and compute the average degree in S. The average degree in S will be an estimate for d.

Estimating the average degree in a graph

Results

Theorem: [Uriel Feige, 2006] Using $O(\epsilon^{-1}\sqrt{n})$ degree queries it is possible to approximate the average degree within $2+\epsilon$ factor with high probability assuming the minimum degree is at least 1.

Note: *With high probability* means with probability at least $1 - n^{-c}$ for some constant c.

Note: The algorithm outputs d' where $(\frac{1}{2} - \epsilon)d \leq d' \leq (1+\epsilon)d$.

This lecture: We prove a similar but weaker result.

Reference: Artur Czumaj, Christian Sohler. Sublinear time algorithms (draft). Available at Artur Czumaj's webpage.

Estimating the average degree in a graph

Analysis

We have a sequence of n (unknown) integers between 1 and $n-1$

$$d_1,\ d_2,\ d_3,\ d_4,\ \ldots, d_n$$

d_i is the degree of i-th node in the graph G.

$$d = \frac{d_1 + d_2 + \ldots + d_n}{n}$$

We sample s nodes (with replacement) and output their average degree. $|S| = s$

Let X_i be a random variable associated with the degree of i-th node in the sampled set S.

Estimating the average degree in a graph

Analysis

Output of the algorithm: $X = \frac{1}{s}(X_1 + \ldots X_s)$

$$E[X_i] = \sum_{i=1}^{n}(d_i \times \frac{1}{n}) = d$$

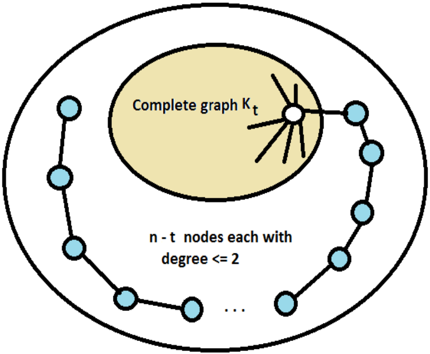
$$E[X] = E[\frac{1}{s}(X_1 + \ldots X_s)] = \frac{1}{s}\sum_{i}^{s} E[X_i] = \frac{1}{s}ds = d$$

(By linearity of expectation.)

In expectation, it is all good (the estimator is unbiased) but how often X is close to $E[X]$?

Estimating the average degree in a graph

Analysis: A bad example



$m = t(t-1) + 2(n-t) - 1$

$d = \frac{2m}{n} = \Theta(\frac{t^2}{n})$

$t = n^{2/3} \Rightarrow d = \Theta(n^{1/3})$

If we sample a small set of vertices, with high probability we pick only the blue vertices.

$Pr[\text{ Picking only blue vertices }] =$

$(1 - t/n)^s \approx 1$ when $s = o(n^{1/3})$

The estimated average degree will be $O(1)$.

Main Question: How many samples do we need?

Estimating the average degree in a graph

Analysis: Markov Inequality

Using Markov Inequality, we can easily show that the probability of X overestimating (by large) is small.

Markov Inequality: For every positive random variable X and $a > 0$, we have

$$Pr[X \geq a] \leq \frac{E[X]}{a}$$

In other words,

$$Pr[X \geq aE[X]] \leq \frac{1}{a}$$

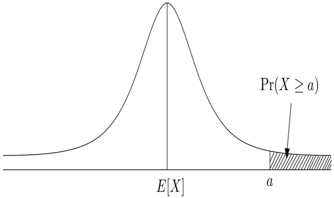
Estimating the average degree in a graph

Analysis: Markov Inequality

Markov Inequality: For every positive random variable X and $a > 0$, we have

$$Pr[X \geq a] \leq \frac{E[X]}{a}$$

$$\begin{aligned} E[X] &= \sum_x xP(x) \\ &= \sum_{x<a} xP(x) + \sum_{x\geq a} xP(x) \\ &\geq \sum_{x\geq a} xP(x) \\ &\geq \sum_{x\geq a} aP(x) \\ &= a\sum_{x\geq a} P(x) \\ &= aP(x \geq a), \end{aligned}$$



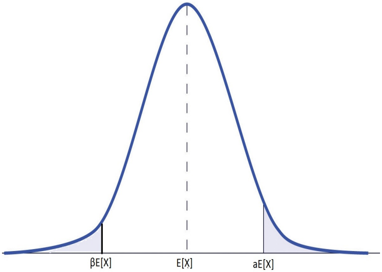
Estimating the average degree in a graph

Analysis: probability of underestimating

Recall that $X = \frac{1}{s}(X_1 + \ldots + X_s)$

Question: Assuming β is a small constant, how large s should be so that we have the following?

$$Pr[X \leq \beta E[X]] = \text{ small}$$



Estimating the average degree in a graph

Analysis: probability of underestimating

Choosing $s \geq \Omega(\epsilon^{-1}\sqrt{n})$ is good enough.

We need to introduce Hoeffding inequality which concerns the analysis of sum of independent variables.

$$X_1 + X_2 + \ldots + X_t$$

X_i's are independent.