Lecture 2: Introduction to sublinear algorithms

Themis Gouleakis

April 20, 2021

Sublinear-time algorithms (examples)

Problem: Compute the diameter of a point set

- m points in metric space X.
- Distances given by:

$$D = \begin{bmatrix} 0 & d_{12} & \dots & d_{1m} \\ d_{21} & 0 & & \vdots \\ \vdots & & \ddots & \\ d_{m1} & \dots & & 0 \end{bmatrix}$$

  - Symmetric: $d_{ij} = d_{ji}$
  - Triangle inequality: $d_{ij} < d_{ik} + d_{kj}$
- Input size: $n = \Theta(m^2)$

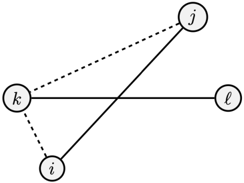
Algorithm 1: Diameter-Estimator

Input : m points in a metric space, matrix $\mathbf{D}$ of all the pairwise distances.
Output: Points k, ℓ and distance $\mathbf{D}_{k,\ell}$.

1 Pick k arbitrarily from $\{1, \cdots, m\}$;
2 Pick ℓ that maximizes $\mathbf{D}_{k,\ell}$;
3 Return $k, \ell, \mathbf{D}_{k,\ell}$

Theorem

Diameter-Estimator returns a 2-approximation to the actual diameter.



Sublinear-time algorithms (examples)

Problem: Number of connected components

- Input: $G = (V, E)$, $|V| = n$
- Goal: Estimate $c = \sharp$ connected components of G.

Let n_v : $\sharp$ of nodes in the connected component of v.

- We need to estimate: $c = \sum_{v \in V} \frac{1}{n_v}$

Lemma

For all $v \in V$, it holds that $\frac{1}{\hat{n}_v} - \frac{1}{n_v} \leq \epsilon/2$, where $\hat{n}_v = \min\{n_v, 2/\epsilon\}$

Algorithm

Algorithm 2: $\widehat{n}_v$**-Calculator**

Input : Graph G, vertex v, ϵ
Output: $\widehat{n}_v$.

1. Initialize Breadth-first search (BFS) from v ;
2. **while** *# of unique visited nodes by BFS is* $< \frac{2}{\epsilon}$ **do**
3.   Continue BFS ;
4.   **if** *BFS finishes* **then**
5.   Return number of visited nodes and abort
6. Return $\frac{2}{\epsilon}$

Algorithm 3: $\tilde{c}$**-Calculator**

Input : Graph G, ϵ, b
Output: $\tilde{c}$.

1. $r \leftarrow b/\epsilon^3$;
2. Sample r vertices $v_1, \cdots, v_r$ from G uniformly with replacement ;
3. Compute $\widehat{n}_{v_i}$ for all $1 \leq i \leq r$ using $\widehat{n}_v$**-Calculator** ;
4. Return $\tilde{c} = \frac{n}{r}\left(\sum_{i=1}^{r} 1/\widehat{n}_{v_i}\right)$

Lemma

It holds that: $\Pr[|\hat{c} - \tilde{c}| > \epsilon n/2] \leq 1/4$

Finishing the proof

Theorem

Let c be the number of connected components of G and let $\tilde{c}$ be the output of Algorithm 3. Then, $\Pr[|c - \tilde{c}| \leq \epsilon n] \geq 3/4$.

Proof:

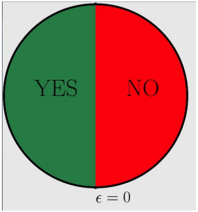
Property testing definitions

Computational problems (**exact**)

Search problems

- $x : R(x) = \{y : (x, y) \in R\}$
- $v : \{0, 1\}^* \to \mathbb{R}$ (value)
- Goal: Find $y^* = \max_{y \in R}\{v(y)\}$

Decision problems



Property testing definitions

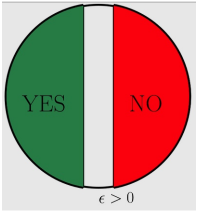
Computational problems (**approximate**)

Search problems

- $x : R(x) = \{y : (x, y) \in R\}$
- $v : \{0, 1\}^* \to \mathbb{R}$ (value)
- Goal: Find

$$y^* : v(y^*) > C \cdot \max_{y \in R(x)} \{v(y)\}$$

Decision problems



Definitions for property testing

Definition

Let Π_n be a set of functions $f : [n] \to R_n$, $n \in \mathbb{N}$. The union $\Pi = \cup_{n \in \mathbb{N}} \Pi_n$ π of these sets will be called a property.

- Oracle access: Query $i \to f(i)$
- Distance: Let $\delta(f, g) = \frac{|\{i \in [n] : f(i) \neq g(i)\}|}{n} = \Pr_{i \in U_{[n]}}[f(i) \neq g(i)]$
- Distance from property $\Pi = \cup_{n \in \mathbb{N}} \Pi_n$:
  - $\delta_\Pi(f) = \delta(f, \Pi) = \min_{g \in \Pi_n}\{\delta(f, g)\}$
  - $\delta_\Pi(f) = \infty$ if $\Pi_n = \emptyset$.
- Query complexity: $q : \mathbb{N} \times (0, 1] \to \mathbb{N}$

Definitions for property testing

Definition

A tester for a property π is a probabilistic oracle machine that outputs a binary verdict that satisfies the following:

1. If $S \in \Pi$, then the tester accepts with probability at least $2/3$.
2. If S is ϵ-far from Π, then the tester accepts with probability at most $1/3$

- **One sided error**: Accept any $S \in \Pi$ with probability 1.

Examples

Problem: Testing convex position

Definition

A point set P is in convex position if every point in P belongs to the convex hull of P.

Definition

A set P of n points is ϵ-far from convex position if no set Q of size (at most) ϵn exists such that $P \setminus Q$ is in convex position.

Goal: Design a tester that can distinguish the above in sub-linear time.

Tester

CONVEXTESTER (P, ϵ)

let $s = 16\left(4\sqrt[d+1]{n^d/\epsilon} + 2d + 2\right)$

Choose a set $S \subseteq P$ of size s uniformly at random

if S is in convex position **then** *accept*

else *reject*

Completeness case: Clearly, if P is in convex position, then any $S \subseteq P$ will also be in convex position.

Soundness case: We need to show that CONVEX TESTER rejects every point set that is ϵ-far from convex position with probability at least $2/3$.