A Sketch-Based Distance Oracle for Web-Scale Graphs

Atish Das Sarma (Georgia Tech.), Sreenivas Gollapudi, Marc Najork, Rina Panigrahy (Microsoft Research)
Friend path on Facebook

Distance/shortest path between YOU and OBAMA?

Too expensive to compute at query time.

300M nodes
10B edges

Could take a day to compute

Need Distance estimate very quickly!
Motivation

• Online Distance Computation – on Massive Graphs
  – Distance/path computation on Social Networks
  – Similarity/Relatedness of URLs on the web
  – Building block for other online algorithms

• Road Networks
  – Already solved very efficiently – specific to 2D

• Same question on web graphs
  – Guarantees weaker, but more general solutions
Previous Approaches - Dijkstra

Dijkstra's Algorithm

Prohibitively expensive at query time, even if parallelized.

Exact Offline Distance Computation

Breadth-First Search

Prohibitively expensive at query time, even if parallelized.
Metric Embeddings

[Bourgain]

Embed into low-dimensional space

Compute the actual distance here

$v \rightarrow (v_1, v_2, \ldots, v_d)$

$u \rightarrow (u_1, u_2, \ldots, u_d)$
Spanner Construction

[Peleg-Schaffer]

Compact Representation but distance still needs to be computed.
Sketch-based

[Thorup-Zwick]

For all nodes $x$

Pre-compute small information

At query time combine

$\text{Sketch}(x)$

$\text{Sketch}(u)$

$\text{Sketch}(v)$

Distance estimated

Metric Embeddings can be thought of as Sketch-based
Problem Definition

**PRECOMPUTATION:**

Preprocess and Store some summary (space about the number of vertices)

At query time, receive $u, v$

**ONLINE:**

Quickly estimate the distance $d(u, v)$
Results (Undirected Graphs)

• Sketch-based algorithm of Thorup-Zwick:
  – Space \( O(\log n) \) per node.
  – Query Time \( O(\log n) \)
  – Distance Approximation (UB) \( (2 \log n - 1) \)

• Metric Embedding of Bourgain, Matousek
  – Same space and (slightly more) query time
  – Distance Approximation (LB) \( (2 \log n - 1) \)
Results (Our Contributions)

• Significant Simplification of Thorup-Zwick
  – Simpler proof of same bounds for simplified algorithm
  \[(2 \log n - 1)\text{--approximation}\]
  – Easy to implement

• Extend algorithms to Directed graphs (without proof)

• Experimental Results
  – Size of preprocessing stored: 480 bytes/node
  – Query Time: \(1.2\) Milliseconds (two disk seeks)
  – Approximation Error
    • Undirected - \(1.2\)
    • Directed - \(1.05\)
Key Technique - Sampling Algorithm (LB)

Bourgain Embedding

Sample random set of Green nodes and store distances from all nodes to the set.

A lower bound on

\[ d(u, v) \geq |d(u, S) - d(v, S)| \]

\[ d(u, S) = \min_{w \in S} d(u, w) \]
Key Technique - Sampling Algorithm (UB)

Idea in Thorup-Zwick

Sample random set of nodes and store nearest node and distance to it from all nodes in the graph.

An upper bound on $d(u, v)$

$$d(u, v) \leq d(u, s) + d(v, s)$$

Since this is true for any seed set $S$, ideal if nearest in seed set is common to both.
Sparse Sampling

Idea in Thorup-Zwick

Upper Bound may be too large

Path may be too long

\[ d(u, s) + d(v, s) \]
Dense Sampling

Idea in Thorup-Zwick

Maybe no common seed

Not an upper bound

Therefore, need sampled set of “correct” size.
Offline Sketch

\[ |S_r| = 2^r \]
\[ |S_{r-1}| = 2^{r-1} \]

\[ r = \lfloor \log n \rfloor \]
Sketches

$|S_r| = 2^r$

$|S_{r-1}| = 2^{r-1}$

$d(u, S_{r-1}) = d(u, u_{r-1})$

$r = \lfloor \log n \rfloor$

$|S_2| = 2^2$

$|S_1| = 2$

$|S_0| = 1$

Repeat a certain number of times
Algorithm (Common Seed)

\[ \min \{d(u, u_t) + d(v, v_t)\} \]

(Where \( u_t = v_t \))
Algorithm

- Pre-computation: All Sketches known.
- Query Time: $u, v$
- Online: Retrieve

$\text{Sketch}(u) \supseteq \{(u_0, \delta^u_0), (u_1, \delta^u_1), \ldots, (u_r, \delta^u_r)\}$
$\text{Sketch}(v) \supseteq \{(v_0, \delta^v_0), (v_1, \delta^v_1), \ldots, (v_r, \delta^v_r)\}$

- Find all $t$ such that $u_t = v_t$

- Set

$$\tilde{d}(u, v) = \min_t \{\delta^u_t + \delta^v_t\}$$
Theorem (similar to Thorup-Zwick)

For Undirected graphs:

\[ d(u, v) \leq \tilde{d}(u, v) \leq (2r - 1)d(u, v) \]

Denote \( d(u, v) \) by \( d \)

Later extend to Directed graphs.
No provable theoretical guarantee
Proof (Undirected)

• Consider balls of radius $d_i$

If seed set such that only one point in it from $A_i \cup B_i$ which is also in $A_i \cap B_i$

Then this point will be in sketch of both $u$ and $v$

It follows,

$$\tilde{d}(u, v) \leq 2d_i$$

$u, v$ in each others’ ball but drawn this way for convenience.
Proof (Undirected)

- Consider balls of radius $d_i$

If $$\frac{|A_i \cap B_i|}{|A_i \cup B_i|} \geq \frac{1}{2}$$

Then with constant probability there exists seed set $S$ such that:

$$S \cap (A_i \cup B_i) = s$$
$$S \cap (A_i \cap B_i) = s$$

It follows with constant probability, $\tilde{d}(u, v) \leq 2d_i$

This can be made with high probability since each size set selected multiple times.
Proof (Continued)

Only remains to show that for some $1 \leq i \leq \log n$:

$$\frac{|A_i \cap B_i|}{|A_i \cup B_i|} \geq \frac{1}{2}$$

This follows by observing:

$$A_i \cup B_i \subseteq A_{i+1} \cap B_{i+1}$$

Therefore, if $r$ different set sizes:

$$\tilde{d}(u, v) \leq 2di \leq 2rd$$

Analysis can be tightened to make it

$$(2r - 1)d$$

Sketch Space:

$$O(\frac{1}{r}n^{1+\frac{1}{r}})$$

Distance approx:

$$(2r - 1)$$

The space-approximation parameter can be traded off
Theorem (Bourgain-Matousek)

• Same seed sets as before.
• For each node $u$, and each seed set $S$ store:
  – $d(u, S)$ (nearest node in set not required)
• Output:
  $$\tilde{d}(u, v) = \max_S (d(u, S) - d(v, S))$$
• Theorem:
  $$\frac{d(u, v)}{(2 \log n - 1)} \leq \tilde{d}(u, v) \leq d(u, v)$$

Again the approximate-space parameter can be traded off.

(Upper Bound follows from Triangle Inequality)
Extending Algorithms to Directed

• Store distances and nearest nodes separately for: \( d(u, S) \) and \( d(S, u) \)

• For estimating \( d(u, v) \) use: \( d(u, S) \) and \( d(S, v) \)

• Theorems do not hold
  – Distances not symmetric.
Experimental Setup

• Web Crawl:
  – 65M webpages, 420M URLs
  – 2.3B edges

• Undirected Distance [1,15]

• Directed Distance
  – Infinite
  – [1,100]

• Sample nodes for evaluation (find pairs from different distances)
Optimization

• Ignore nodes with zero indegree/outdegree

• Hash seed sets identifiers:
  – Lossy compression but saves space
  – Small error

• Sketch size: \((s + 8)k \log n\)
  – \(k = 3\) number of copies of seed sets
  – \(s = 12\) size of seed id. 8 bits to store distance.
  – 240, 480 bytes for undirected, directed.
Evaluation Results - Undirected k=1
Directed $k=1$
Undirected vary k.

The graph shows the estimated distance/true distance ratio for different methods as a function of $k$. The methods include:

- Ideal
- Mean Common-Seed
- 1st quartile Common-Seed
- 2nd quartile Common-Seed
- 3rd quartile Common-Seed
- Mean Bourgain
- 1st quartile Bourgain
- 2nd quartile Bourgain
- 3rd quartile Bourgain

The x-axis represents the value of $k$, ranging from 1 to 20, while the y-axis represents the estimated distance/true distance ratio, ranging from 0 to 2.
Questions

• Directed graphs have lower bound (no sketch-based algorithm can give reasonable distance estimate)

• Why does our algo perform well on the web graph?
  – Additional structure? (sparsity, special connectivity...?)
Thank You!