Landmark Selection Strategies for Computing Distances in Real-World Graphs

Frank Takes

Leiden University, The Netherlands

LIACS Data Mining Workshop — May 15, 2014
Figure: Real-world graph with 1422 nodes and 3711 undirected edges
Graphs are everywhere: (online) social networks, collaboration networks, webgraphs, communication networks, etc.

- Real-world graphs can be large: a graph with 8 million nodes and 1 billion links is not uncommon

- Graphs have all kinds of interesting structural properties: distance distribution, radius, diameter, etc.
  - Common underlying computational task: find a shortest path (compute the distance) from $A$ to $B$
Figure: Distance distribution of an online social network with 8 million nodes and 1 billion links, sampled over 100,000 node pairs.
Distance queries

- Unweighted Graph $G = (V, E)$ with:
  - $|V| = n$ nodes
  - $|E| = m$ links

- **Distance query**: Given nodes $u, v \in V$, compute $d(u, v)$

- Exact approach: Breadth First Search (BFS)

- One BFS takes $O(m)$ time — 6 seconds if $m = 1$ billion

- **Research question**: How to accurately answer thousands of distance queries in a second?
  Some (short) precomputation time will be given.
Landmark framework

Precomputation phase

- Landmark set $L \subseteq V$ with $|L| \ll |V|$
- Common landmark count: $|L| = 100$
- Precompute for all $u \in L$ and $v \in V$ the value of $d(u, v)$
  So, perform 100 full BFSes and store the result.

Query phase: for distance query $d(v, w)$, with $v, w \in V$, return:

- 0 if $v = w$ $O(1)$
- 1 if $w \in N(v)$ $O(\log(m/n))$
- 2 if $N(v) \cap N'(w) \neq \emptyset$ $O(m/n)$
- $\min_{u \in L} (d(v, u) + d(u, w))$ otherwise $O(|L|)$

(here, $N(v)$ and $N'(v)$ are sets of nodes with a link from resp. to $v$)
Landmark framework example (0)

Landmarks $F$ and $P$

Precompute for all $v \in V$: $d(F, v)$ and $d(P, v)$
Compute $d(A, C)$

$C \in N(A)$, so $d(A, C) = 1$
Landmark framework example (2)

Compute $d(B, D)$

$N(B) \cap N(D) \neq \emptyset$, so $d(B, D) = 2$
Compute $d(F, Q)$ via $F$: $d(F, F) + d(F, Q) = 0 + 4 = 4$
Landmark framework example (4)

Compute $d(E, N)$

via $P$: $d(E, P) + d(P, N) = 4 + 1 = 5$

via $F$: $d(E, F) + d(F, N) = 1 + 3 = 4$
Landmark selection

- Deciding if some set of landmarks is optimal, is NP-hard
- Baseline: a random landmark set $L$ from the node set $V$
- Better: select the top-$|L|$ nodes from the node list, sorted using some centrality measure:
  - Degree centrality
  - PageRank centrality
  - Betweenness centrality
  - **Adaptive selection**: select nodes that contribute the most to reducing the error rate
- Even better: process the list of nodes in a smart way
  - Skip direct neighbors
  - **Greedy central neighbor (gcn) processing**: select $h > 0$ times a selected node’s most central neighbor (according to some centrality measure), if such a neighbor exists
Comparison of landmark strategies

- CA-HepPh network:
  - Scientific collaboration network, field of high energy physics
  - $n = 11200$ nodes
  - $m = 235000$ edges
  - Average node-node distance of 4.66

- Compute for each landmark selection strategy the **node error** as follows:

$$\frac{|d_{real} - d_{estimate}|}{d_{real}}$$
Node errors based on degree centrality
Node errors based on gc
Conclusion & Future work

- Shortest path lengths in real-world graphs can accurately be approximated using the landmark framework (error < 0.05)
- The final result can be optimized using various shortcutting techniques
- A good landmark selection strategy takes into account at least two things:
  - Selecting central nodes
  - Covering different areas of the graph
- Finding an optimal set of landmarks will (probably) always remain a challenging task