Fast Shortest Path Distance Estimation in Large Networks

Michalis Potamias

Francesco Bonchi

Carlos Castillo

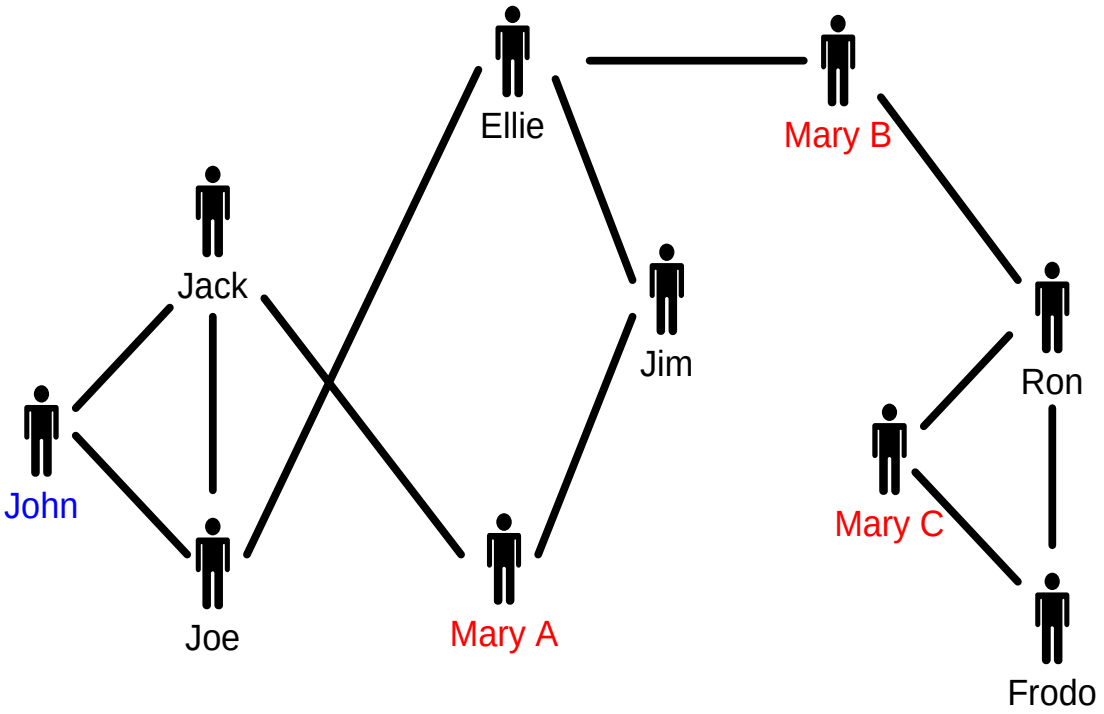
Aristides Gionis

YAHOO! RESEARCH

Context-aware Search

…use shortest-path distance in wikipedia links-graph!

Social Search



John searches Mary

Ranking:

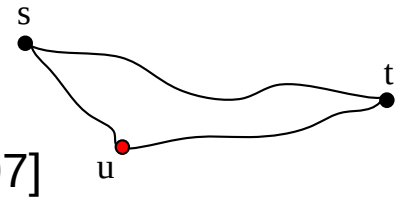
1. Mary A
2. Mary B
3. Mary C

…use shortest-path distance in friendship graph!

Problem and Solutions

- DB: Graph $G = (V,E)$
- Query: Nodes s and t in V
- Goal: Compute fast shortest path $d(s,t)$

- Exact Solution
  - BFS - Dijkstra
  - Bidirectional - Dijkstra with A* (aka ALT methods)
    - [Ikeda, 1994] [Pohl, 1971] [Goldberg and Harrelson, SODA 2005]
- Heuristic Solution
  - Avoid traversals – Use Random Landmarks
    - [Kleinberg et al, FOCS 2004] [Vieira et al, CIKM 2007]
  - Can we choose Better Landmarks ?!?



The Landmarks' Method

- Offline
  - Precompute distance of all nodes to a small set of nodes (landmarks)
  - Each node is associated with a vector with its SP-distance from each landmark (embedding)
- Query-time
  - $d(s,t)$ = ?
  - Combine the embeddings of s and t to get an estimate of the query

Contribution

1. Proved that covering the network with landmarks is NP-hard.

2. Devised heuristics for good landmarks.

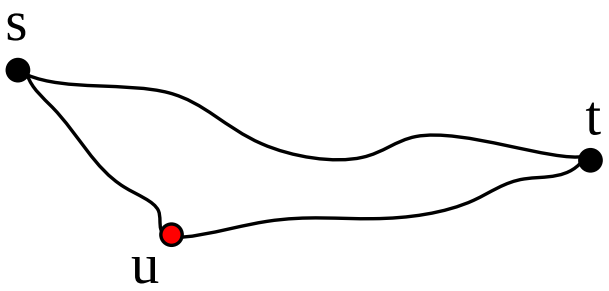
3. Experiments with 5 large real-world networks and more than 30 heuristics. Comparison with state of the art.

4. Application to Social Search.

Algorithmic Framework

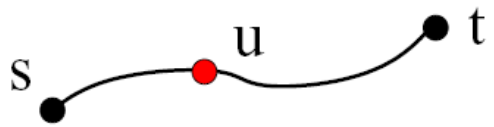
- Triangle Inequality

$$d_G(s,t) \leq d_G(s,u) + d_G(u,t),$$
$$d_G(s,t) \geq |d_G(s,u) - d_G(u,t)|$$



- Observation: the case of equality

$$d_G(s,t) = d_G(s,u) + d_G(u,t)$$



$$d_G(s,t) = |d_G(s,u) - d_G(u,t)|$$

The Landmarks' Method

1. Selection: Select k landmarks

2. Offline: Run k BFS/Dijkstra and store the embeddings of each node:

   $\Phi(s) = \langle d(s, u_1), d(s , u_2), \ldots , d(s, u_k) \rangle$

   $= \langle s_1, s_2, \ldots, s_k \rangle$

3. Query-time: $d(s,t) = ?$
   - Fetch $\Phi(s)$ and $\Phi(t)$
   - Compute $\min_i\{s_i + t_i\}$ (i.e. inf of UB) ... in time $O(k)$

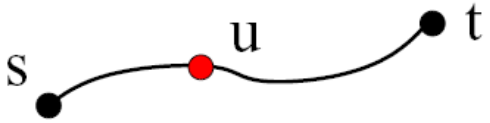
Example query: $d(s,t)$

	$d(\_,u_1)$	$d(\_,u_2)$	$d(\_,u_3)$	$d(\_,u_4)$
$\Phi(s)$	2	4	5	2
$\Phi(t)$	3	5	1	4

UB	**5**	9	6	6
LB	1	1	**4**	2

$$\max_i |s_i - t_i| \leq d_G(s,t) \leq \min_j \{s_j + t_j\}$$

Coverage Using Upper Bounds



- A landmark *u* covers a pair (*s*, *t*), if *u* lies on a shortest path from *s* to *t*
- Problem Definition: find a set of *k* landmarks that cover as many pairs (*s*,*t*) in *V* x *V* as possible
  - NP-hard
  - *k* = 1 : node with the highest betweenness centrality
  - *k* > 1 : greedy set-cover (approximation - too expensive)

…central nodes are a good start for devising heuristics!

Landmarks Selection: Basic Heuristics

- Random (baseline)

- Choose central nodes!
  - Degree
  - Closeness centrality
    - Closeness of *u* is the average distance of *u* to any vertex in *G*

- Caveat: many central nodes may cover the same pairs: newly added landmarks should cover different pairs

…spread the landmarks in the graph!

Constrained Heuristics

- Remove immediate neighborhood

  1. Rank all nodes according to Degree or Centrality
  2. Iteratively choose the highest ranking nodes. Remove *h*-neighbors of each selected node from candidate set

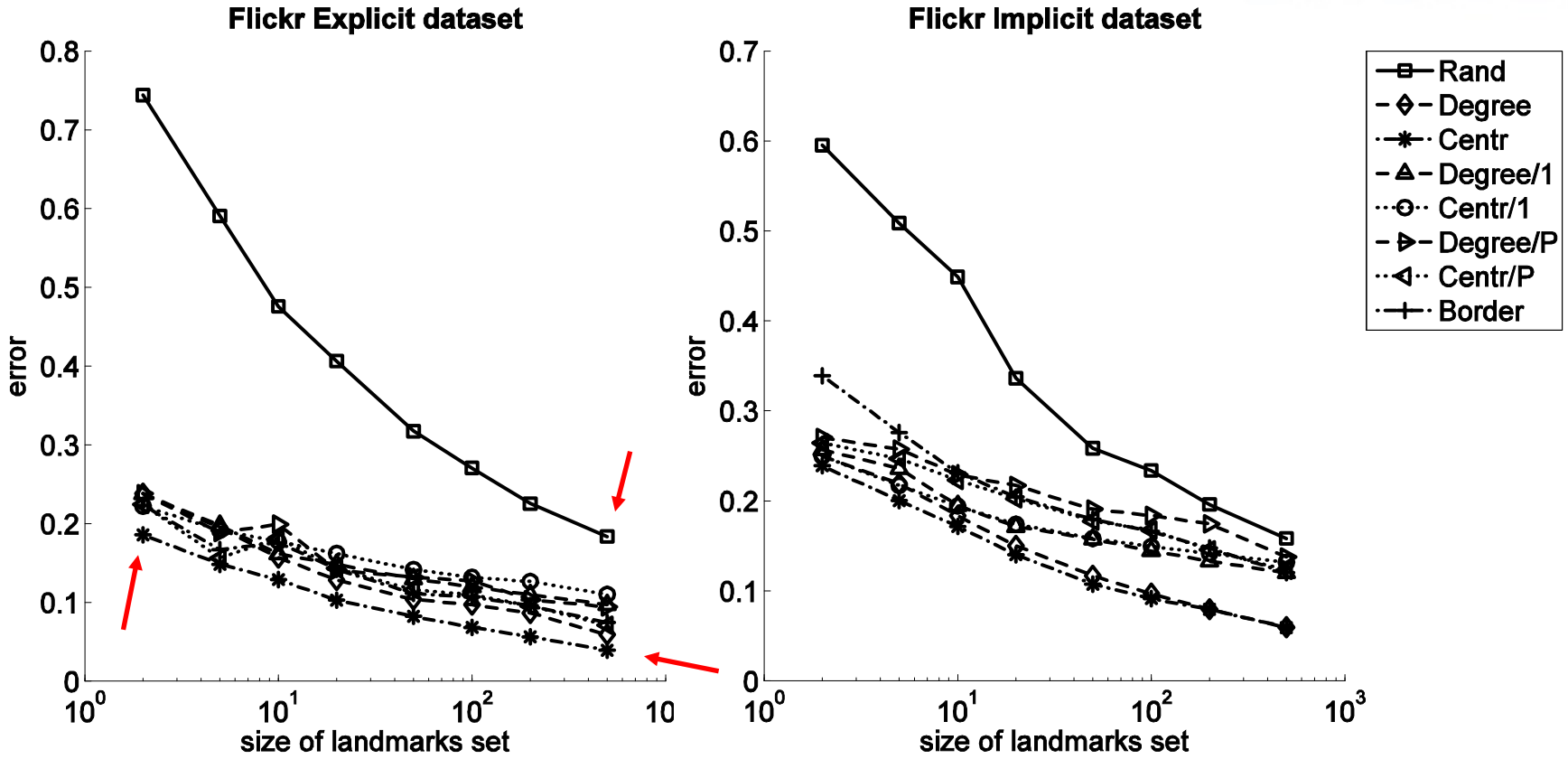
- Denote as
  - Degree/h
  - Closeness/h
  - Best results for $h = 1$

Partitioning-based Heuristics

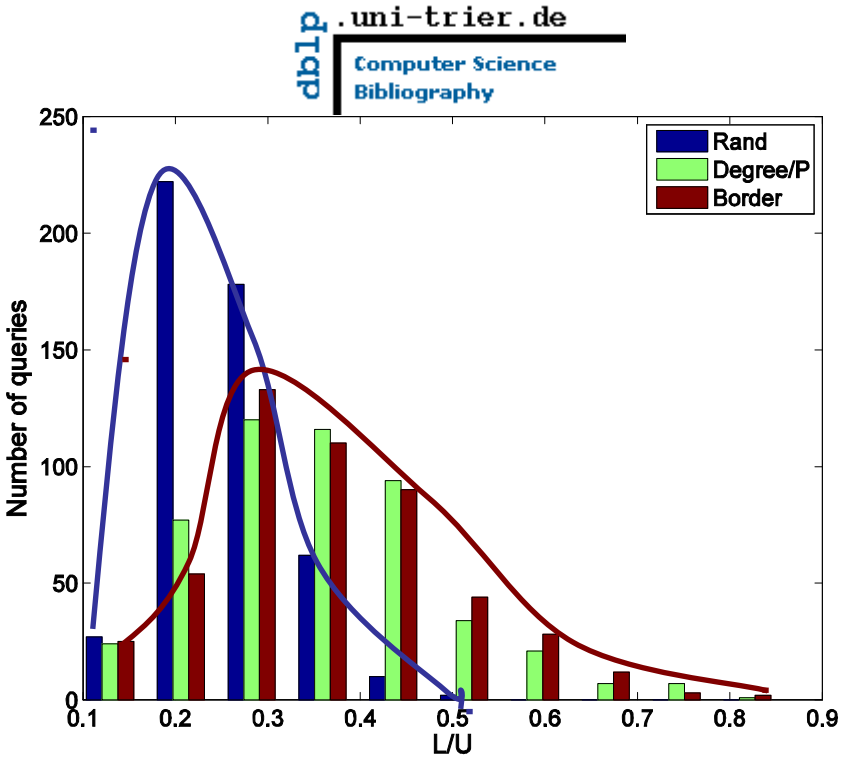
- Use graph-partitioning to spread nodes.
- Utilize any partitioning scheme and
  - Degree/P
    - Pick the node with the highest degree in each partition
  - Closeness/P
    - Pick the node with the highest closeness in each partition
  - Border/P
    - Pick the node closer to the border in each partition. Maximize the border-value that is given from the following formula:

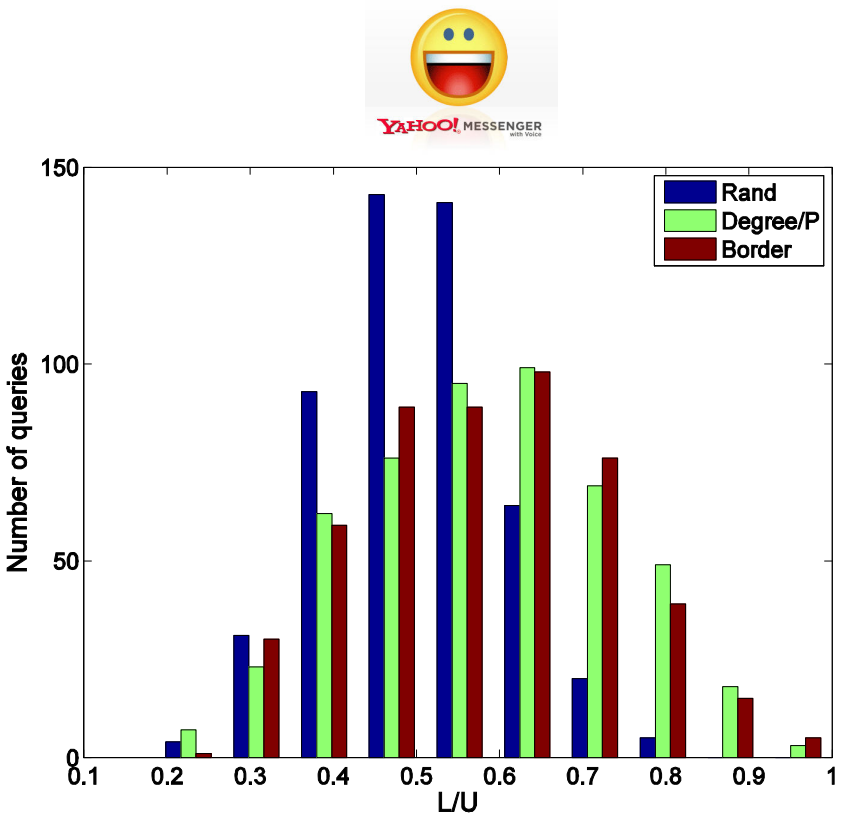
$$b(u) = \sum_{j \in C, u \in C(i), i \neq j} d_j(u) \cdot d_i(u)$$

Versus Random - error



Versus Random - triangulation





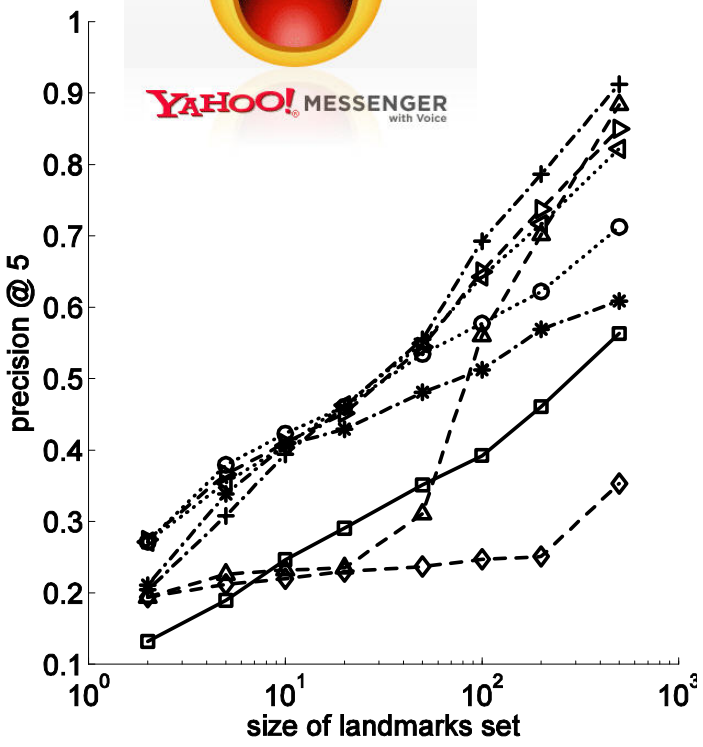
random landmarks have theoretical guarantees [FOCS04]

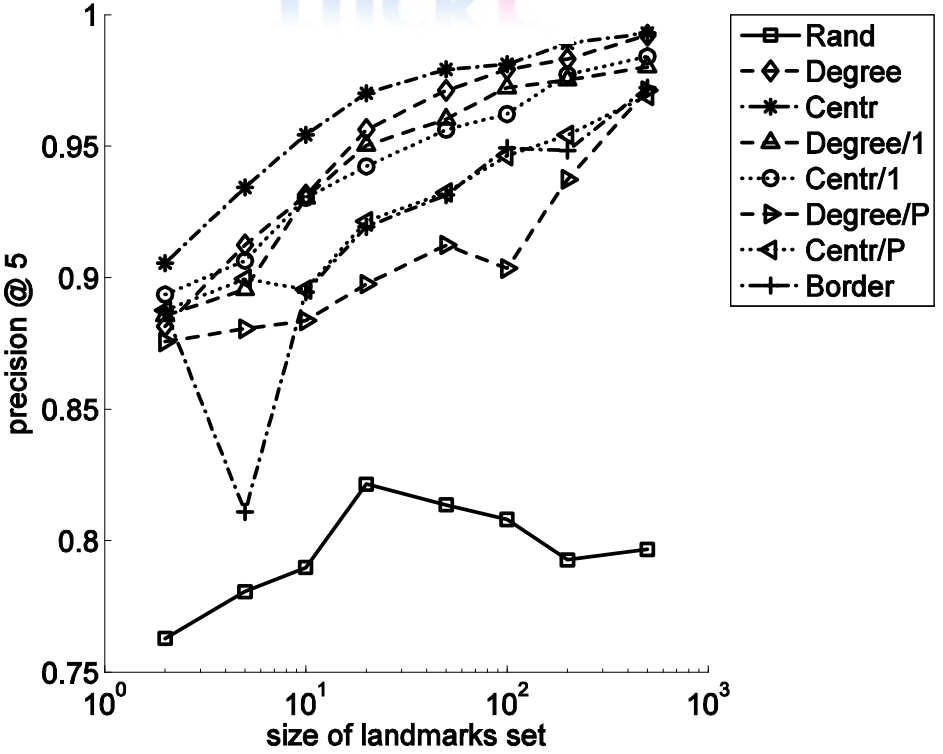
Versus ALT - efficiency

	flickr™	flickr™	WIKIPEDIA	dblp .uni-trier.de Computer Science Bibliography	YAHOO! MESSENGER with Voice
Ours (10%) Operations	20	100	500	50	50
ALT LB Operations	60K >300x	40K >400x	80K >160x	20K >400x	2K >40x
ALT Visited Nodes	7K	10K	20K	2K	2K

state of the art exact ALT methods [SODA05]

Social Search Task





random landmarks have been used [CIKM07]

Conclusion

- Novel search paradigms need distance as primitive
  - Approximations should be computed in milliseconds
- Heuristic landmarks yield remarkable tradeoffs for SP-distance estimation in huge graphs
  - Hard to find the optimal landmarks
  - Border and Centrality heuristics:
    - outperform Random even by a factor of 250.
    - are, for a 10% error, many orders of magnitude faster than state of the art exact algorithms (ALT)
- Future Work
  - Provide fast estimation for more graph primitives!

Thank you!

?