Boosting PageRank Scores by Optimizing Internal Link Structure

Naoto Ohsaka (NEC / UTokyo)
Tomohiro Sonobe (NII)
Naonori Kakimura (Keio University)
Takuro Fukunaga (RIKEN AIP)
Sumio Fujita (Yahoo Japan Corp.)
Ken-ichi Kawarabayashi (NII)

ERATO

Overview of contributions

Q. How to **boost** PageRank by adding a few edges?

[1] Problem formulations

Finding K missing edges s.t. inserting them maximizes the minimum PageRank among T target vertices

[2] Complexity analysis

NP-hard to solve (and even to approximate)

[3] Algorithms

Greedy-based heuristic with
a novel measurement of edges' contribution

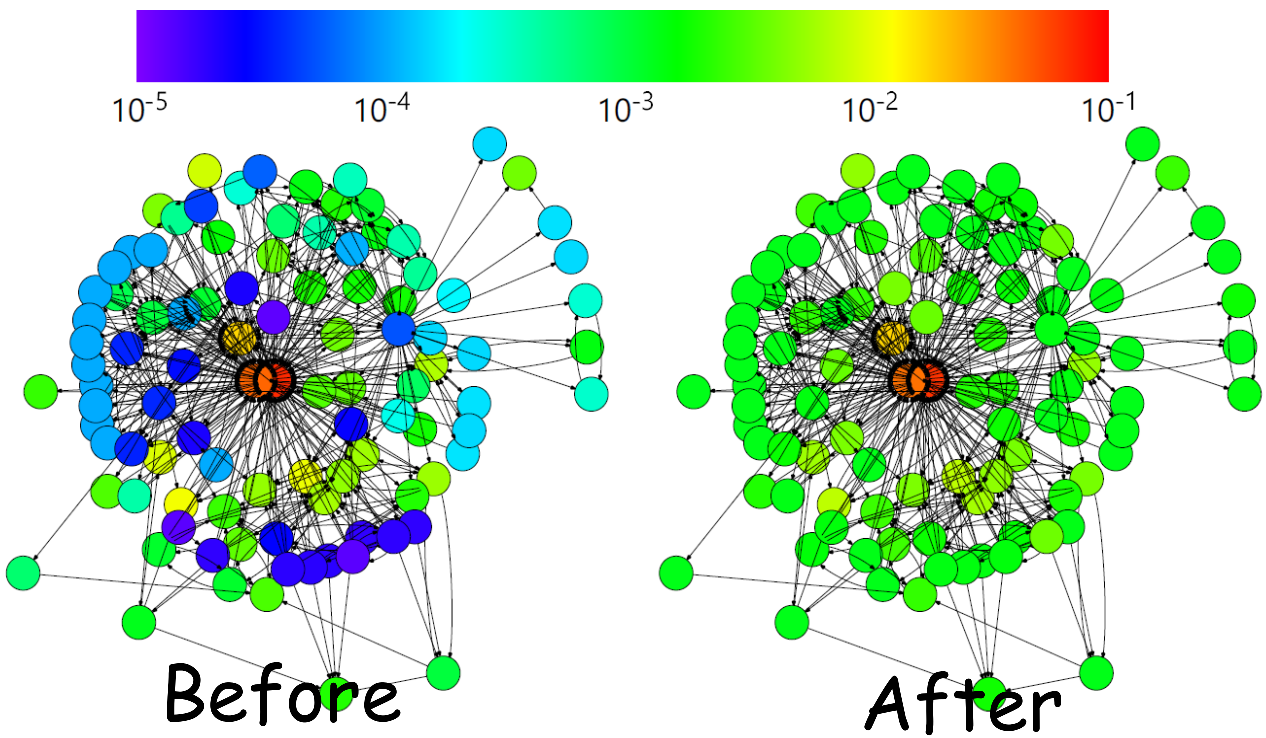
[4] Experimental evaluations

Adding a few edges can increase PageRank scores

Example of experimental results

- Task: inserting **80** edges that maximize the min. PageRank among **100** vertices

Subgraph induced by the 100 vertices
(colored according to PageRank scores)



Google's notion **PageRank (PR)**

[Brin-Page. *1998*] [Page-Brin-Motwani-Winograd. *1999*]

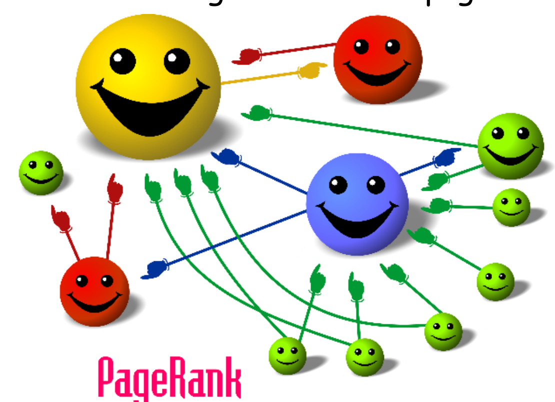
Measures the importance of webpages based on the structure of graph $G = (V, E)$

$$\mathbf{x} = \alpha P \mathbf{x} + (1-\alpha)|V|^{-1}\,\mathbf{1}$$

Transition matrix (P)

Decay factor = 0.85 (α)

https://en.wikipedia.org/wiki/File:PageRank-hi-res.png

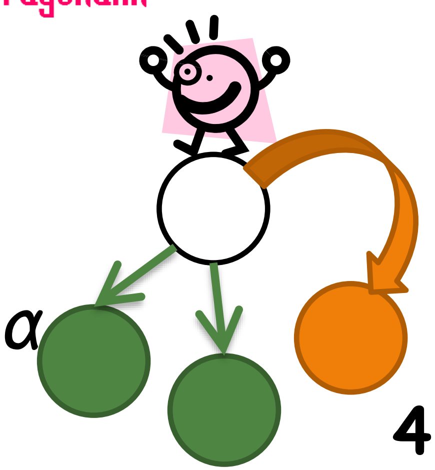


Random-walk interpretation ⇅ *Stationary distribution*

Random walk modeling web browsing

Moves to a random out-neighbor w.p. α

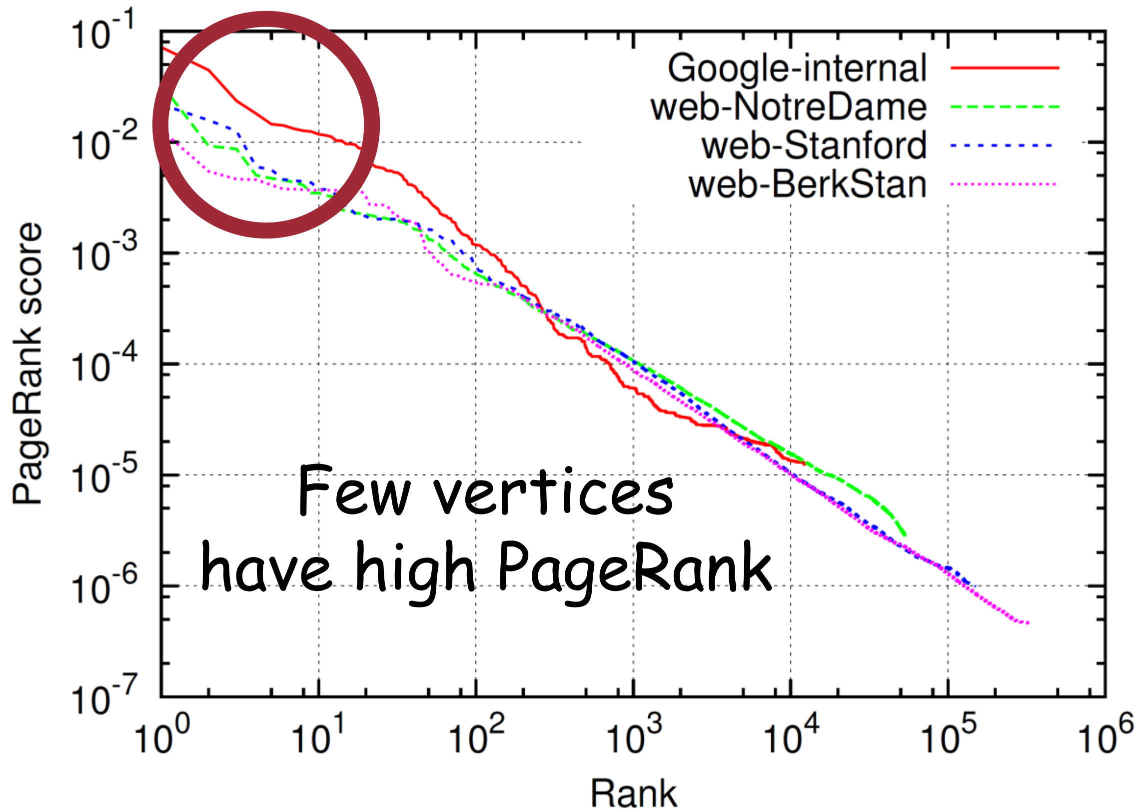
Jumps to a random vertex w.p. $1-\alpha$



Motivation of **boosting** PageRank

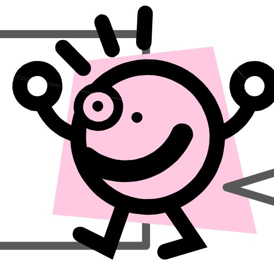
FACT: The distribution of PageRank is a *power-law*

[Becchetti-Castillo. *WWW 2006*]
[Fortunato-Boguñá-Flammini-Menczer. *WAW 2006*]
[Pandurangan-Raghavan-Upfal. *COCOON 2002*]



E.g., Online advertising

Advertiser
Creates ads



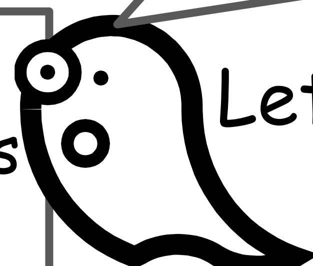
Want to buy banner spaces frequently visited by users

Customer
Views ads

Popularity ≈ PageRank

Want to <u>guide many users</u> to **every** webpages w/ banner spaces

Publisher
Sells banner spaces to advertisers



Let's **optimize** the structure of the host network!!

Problem formulations

PageRank (PR) boosting

$G = (V, E)$: directed graph, $T \subseteq V$: set of targets
K : # missing edges, L : threshold value

MPM(T, K): Minimum PageRank Maximization
Find K edges not in E maximizing the min. PR among T

MinPTC(T, L): Minimum PageRank Threshold Coverage
Find min. # edges not in E s.t. every PR in $T \geq L$

Generalization

MaxPTC(T, K, L): Maximum PageRank Threshold Coverage
Find K edges not in E maximizing (# vertices in T of PR $\geq L$)

Related studies and known results

Outgoing edges from target vertices are allowed

[Avrachenkov-Litvak. *Stoch. Model 2006*] [Sydow. *AWIC 2005*]
[de Kerchove-Ninove-Van Dooren. *Linear Algebra Appl. 2008*]

- Optimal linking (clique-like) structure exists

Incoming edges to target vertices are allowed

[Olsen. *CIAC 2010*] [Olsen-Viglas-Zvedeniouk. *COCOA 2010*]
[Olsen-Viglas. *Theor. Comput. Sci. 2014*]

- Constant approx. is possible if $|T| = 1$

General case (Edges under control are allowed)

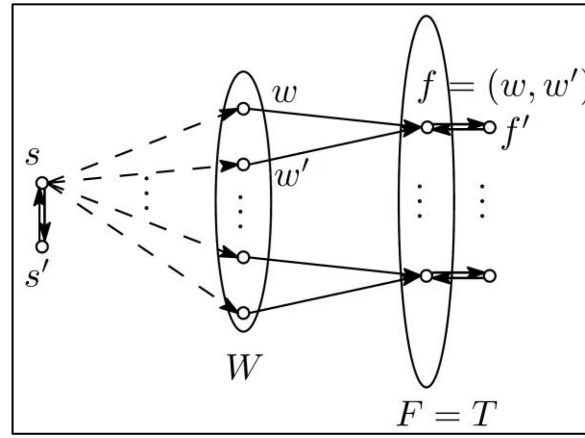
[Olsen. *COCOON 2008*] [Csáji-Jungers-Blondel. *ALT 2010*]
[Csáji-Jungers-Blondel. *Discrete Appl. Math. 2014*]

- Polynomial time if $|T| = 1$
- A variant of **MPM** is NP-hard

The three problems are NP-hard

Vertex Cover → **MPM** on a simple cubic graph

Reduction



MPM → **MinPTC** or **MaxPTC**

Reduction w/ bisection search

KEY: For any G and G' on V,

$x(v) = x'(v)$ or $|x(v) - x'(v)| \geq 1/|V|^{|V|+1}$

Vertex Cover is NP-hard (to approximate < 1.3606)

[Dinur-Safra. *Ann. Math. 2005*]

Idea for **MinPTC** (See paper for other problems)

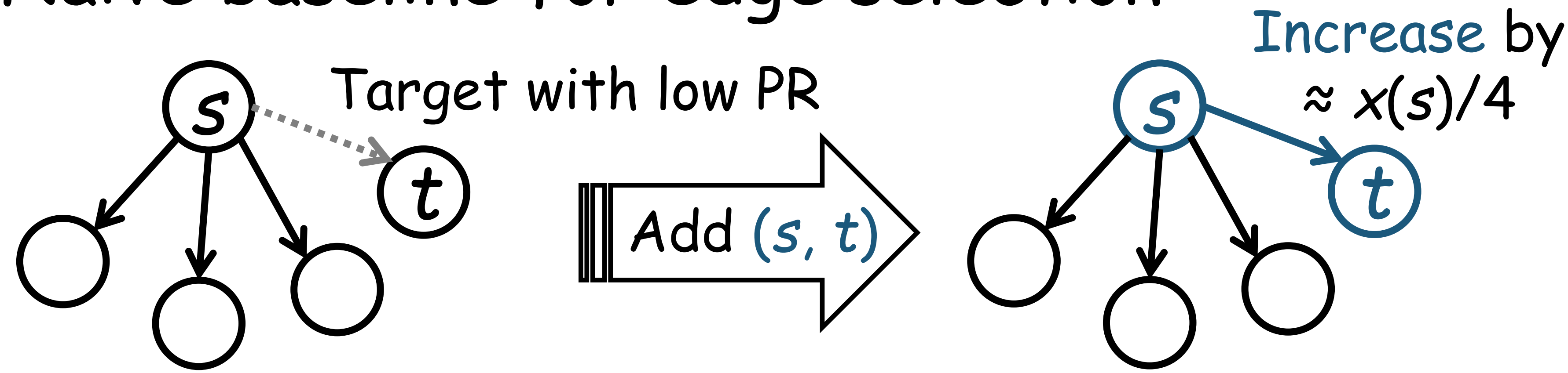
Recall: Find min. # new edges s.t. every PR in $T \geq L$

Use **Greedy Algorithm** ($\because$ Approx. for Vertex Cover)
[Johnson. *J. Comput. Syst. Sci.* 1974]
[Lovász. *Discrete Math.* 1975] [Chvatal. *Math. Oper. Res.* 1979]
Repeatedly add a new edge (s, t) to G until every PR ≥ L

Key insight: adding edge (s, t), then $x(t)$ increases
[Avrachenkov-Litvak. 2004] [Ipsen-Wills. 2006]

Which s (in V) & t (in T) are good ?

Naive baseline for edge selection



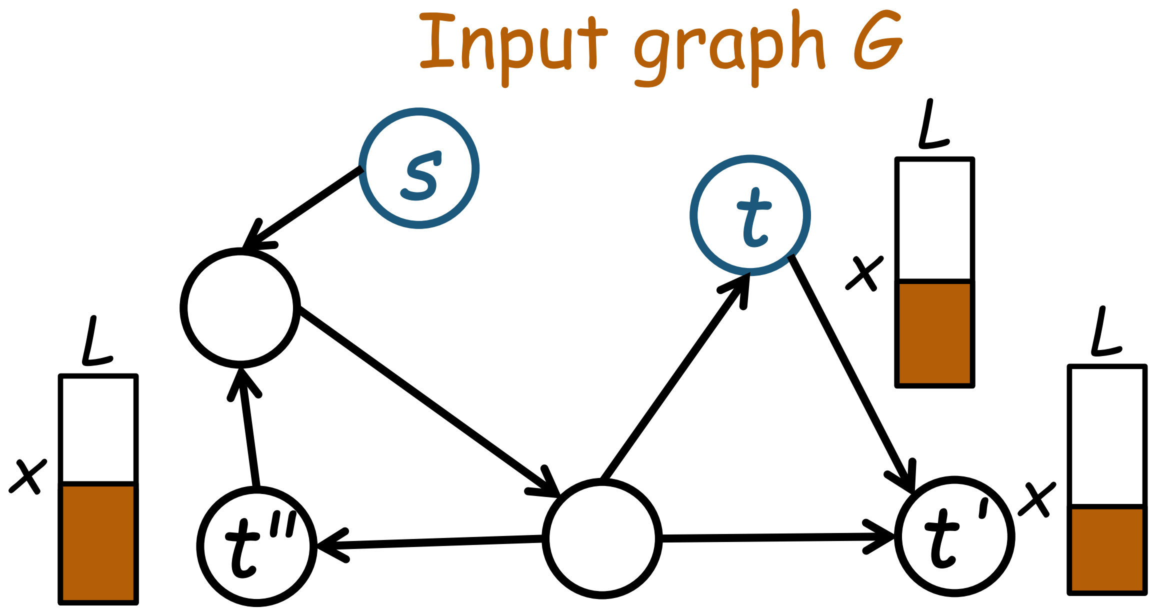
Each iter. returns (s, t) s.t.

- t is a target with min. PageRank
- s is a vertex with max $\frac{x(s)}{d(s)+1}$

NOTE: [Olsen-Viglas-Zvedeniouk. *COCOA 2010*]
[Olsen-Viglas. *Theor. Comput. Sci. 2014*]
proposed this alg. for the case $|T| = 1$

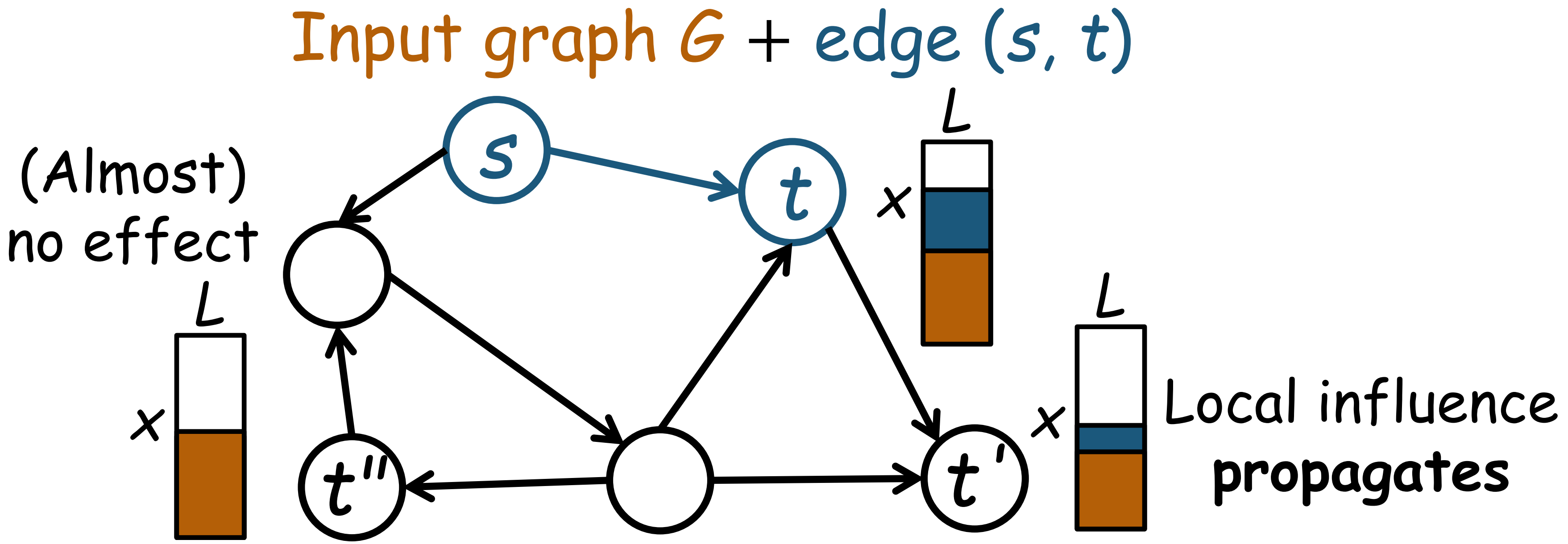
Our approach for edge selection

- Directly measure the value of edge (s, t)



Our approach for edge selection

- Directly measure the value of edge (s, t)



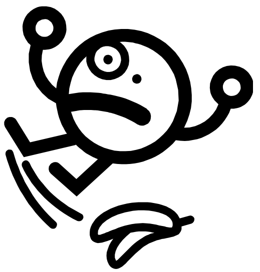
Contribution is

$$\sum_{v \in T} \min\{x(v) - L, 0\}$$

Takes 0 if every PageRank in $T \geq L$

x: PageRank on $G + (s, t)$

Speeding-up techniques



Each greedy iteration computes PageRank on $G + (s, t)$ for **every** candidate (s, t) in $V \times T$?

1. Discard small-PageRank vertices

E.g., s should rank the top-$|T|$

2. Use dynamic (incremental) algorithms

E.g., [O.-Maehara-K. *KDD 2015*]
[Zhang-Lofgren-Goel. *KDD 2016*]

Experimental evaluations

Results on **MinPTC** (See the paper for others)

settings		# inserted edges		run time [s]	
$\lvert T \rvert = 100$	L	proposed	naive	proposed	naive
Random	0.0001	101	**100**	840	3
Random	0.0008	**780**	1,027	4,577	20
2-hop	0.0001	**11**	32	100	1
2-hop	0.0008	**148**	233	1,019	5

Stanford webgraph ($\lvert V \rvert$=**150K**, $\lvert E \rvert$=**1.6M**) [Stanford Network Analysis Project]

Random : random 100 vertices

2-hop: 2-hop neighbors of a high-PR vertex

✓ **Proposed** alg. requires fewer edges than **naive**

✓ **Proposed** alg. scales million-edge networks

Environment: Linux server w/ Intel Xeon E5540 2.53 GHz CPU and 48 GB RAM

Conclusion and future work

4 contributions on PageRank boosting:

[1] Problem formulations

[2] Complexity analysis

[3] Algorithms

[4] Experimental evaluations

- Approx. guarantee

  Constant factor when $|T| = 1$

  [Olsen-Viglas-Zvedeniouk. *COCOA 2010*]

  Difficulty: taking the **minimum** among T
- Further acceleration