Local is Good: A Fast Citation Recommendation Approach

Haofeng Jia and Erik Saule
Department of Computer Science
UNC Charlotte
hjia1@uncc.edu, esaule@uncc.edu

Introduction

Context: Citation recommendation is now a common tool that academics rely one. It can be broadly defined as from *who I am* or *what my interests are* provide relevant papers.
Target Systems: The query of the recommender system is expressed as a set of existing papers. theAdvisor [3] uses explicit query, also passive recommender systems use past publications as a set of query paper.
Problem: There is an inherent trade-off between obtaining a good recommendation and a fast recommendation.

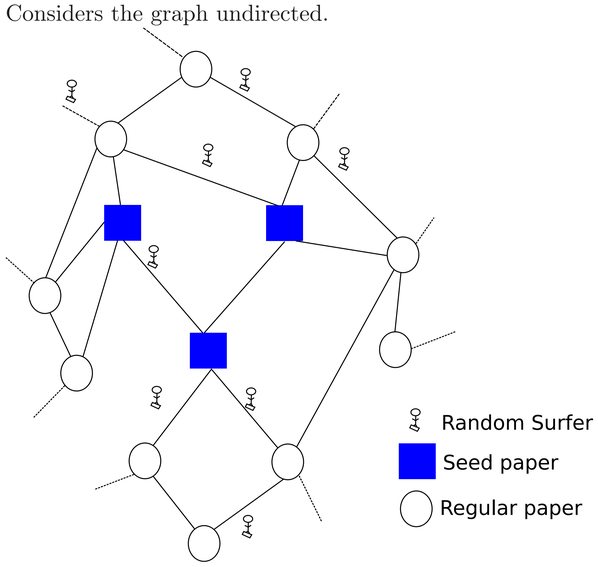
Can we obtain a fast and good recommendation?

Problem Definition

Let $G = (V, E)$ be the citation graph, with n papers $V = \{v_1, \ldots, v_n\}$. Each edge $e \in E$ represents a citation relationship between two papers.
Citation Recommendation. Given a set of seed papers S as a query, return a list of papers ranked by relevance to the ones in S.

PaperRank [1]

Biased random walk where the restarts probability from any paper will be distributed to only the seed papers. Considers the graph undirected.



Collaborative Filtering [2]

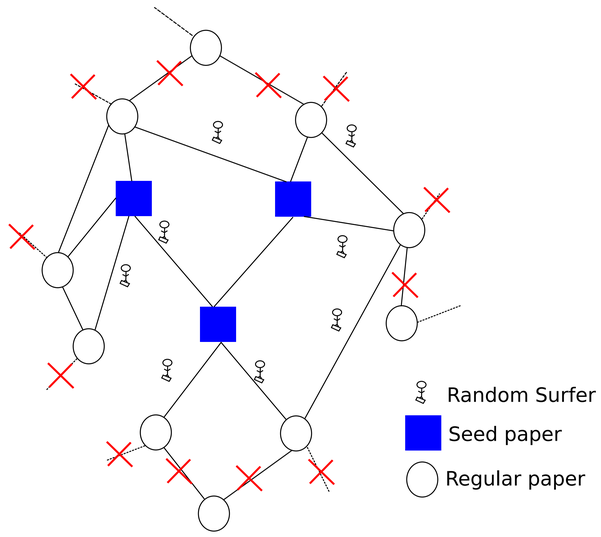
For citation recommendation:

- Build a ratings matrix using the adjacency matrix of the citation graph
  - citing papers correspond to users
  - citations correspond to items.
- Add a pseudo target paper that cites all seed papers
- Computes the cosine similarity of all papers with the target paper
- Identify x *peer* papers, having the highest similarity to the target paper.
- Each paper is scored by summing the similarity of the peer paper that cites it.

LocRank [ThisPaper]

Inspired from PaperRank.
Consider only the ego network of the seed papers.

- Remove all papers at a distance of 2 or more from a seed paper
- Retain all edges between these papers



Experimental setting

Dataset: a corpus of 2M Computer Science papers and 12M citations obtained by mapping :

- Microsoft Academic Graph
- CiteSeerX
- DBLP

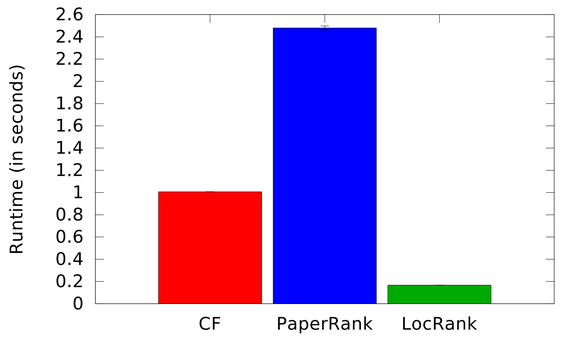
Queries: 2,500 random hide-10% queries:

- Pick randomly a query paper q with 20 to 200 references and published between 2005 to 2010.
- Remove q and papers published after q
- Use a random 90% of the references of q as seed paper S
- Use the remaining 10% as hidden papers to discover

Hardware and Software:

- C++ code compiled with g++ 4.8.2 with -O3.
- Graphs are in Compressed Row Storage format
- Codes are run on 1 core of an Intel(R) Xeon CPU E-5-2623 @ 3.00GHz processor.

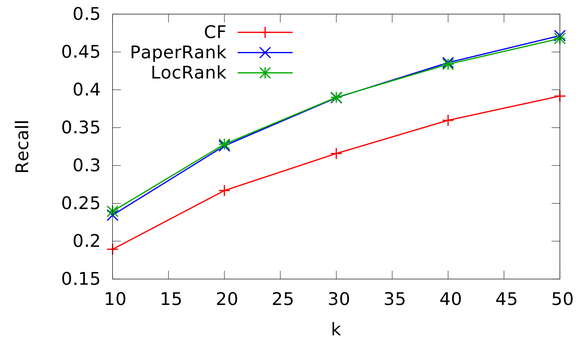
Runtime



LocRank is

- 15x faster than PaperRank
- 6x faster than CF

Recall@k



LocRank

- Recalls more papers than CF
- Recall the same number of paper as PaperRank

Conclusion

LocRank is:

- Faster than both CF and PaperRank
- Higher recall than CF and same as PaperRank

LocRank only needs local context, missing information far from the seed will not impact performance.
LocRank can not find loosely connected papers but they are hard to find in general [4].
How to integrate non-citation information?

References

[1] Gori, M., Pucci, A.: Research paper recommender systems: A random-walk based approach. In: Proc. of Web Intelligence. (2006) 778–781
[2] McNee, S.M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S.K., Rashid, A.M., Konstan, J.A., Riedl, J.: On the recommending of citations for research papers. In: Proc. of CSCW. (2002) 116–125
[3] Küçüktunç, O., Saule, E., Kaya, K., Çatalyürek, Ü.V.: Towards a personalized, scalable, and exploratory academic recommendation service. In: Proc. of ASONAM. (2013)
[4] Jia, H., Saule, E.: An analysis of citation recommender systems: Beyond the obvious. In: Proc of ASONAM. (2017)

Acknowledgments

This work was partially supported by NSF under Grant No. 1652442.