Local Algorithms and Large Scale Graph Mining

Silvio Lattanzi (Google Research NY)

Charles River Workshop on Private Analysis of Social Networks

Outline

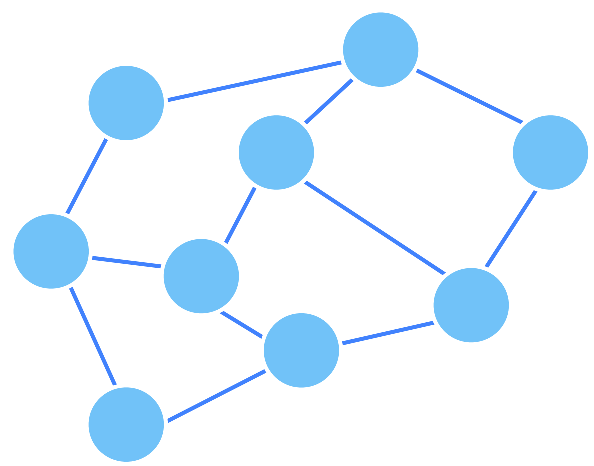
- **Problem and challenges**
  Graph clustering, computation limitations.
- **Local random walk and node similarities**
  Personalize page rank to detect similar nodes in a graph.
- **Local random walk and clustering in practice**
  Personalize page rank and distributed clusters in practice.
- **Local clustering beyond Cheeger's inequality**
  A local algorithm for finding well connected clusters.

Problem and challenges

Local graph algorithms

Local algorithms

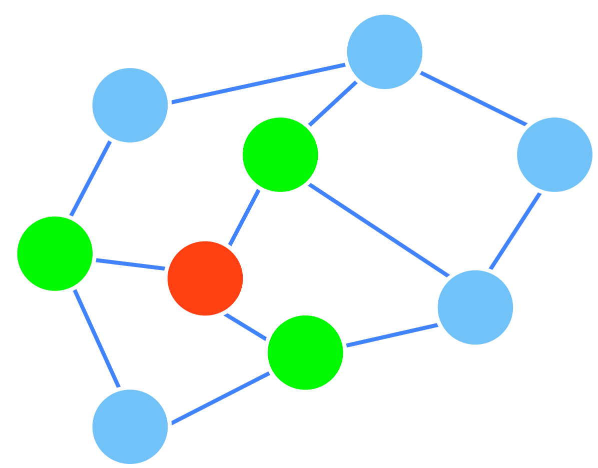
Algorithms based on *local* message passing among nodes



Local graph algorithms

Local algorithms

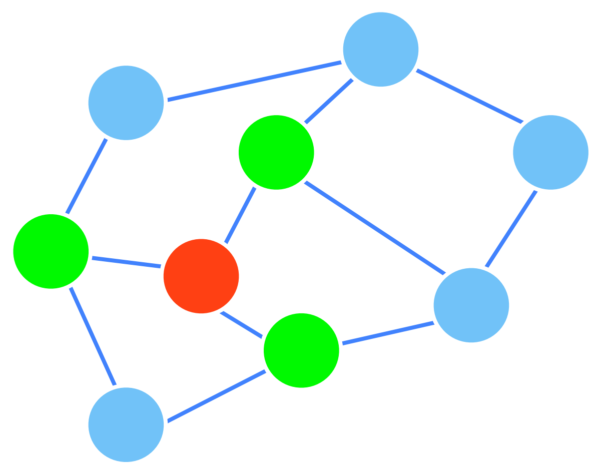
Algorithms based on *local* message passing among nodes



Local graph algorithms

Local algorithms

Algorithms based on *local* message passing among nodes



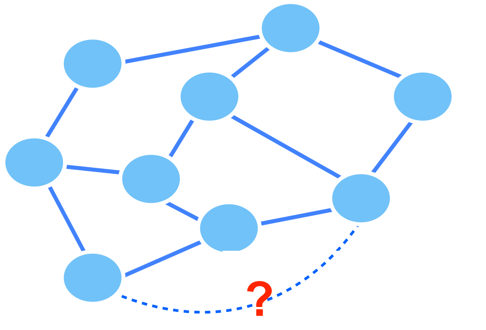
Advantages

- Applicable to large scale graphs
- Fast, easy to implement in parallel (MapReduce, Hadoop, Pregel...)

Problems

Similarity

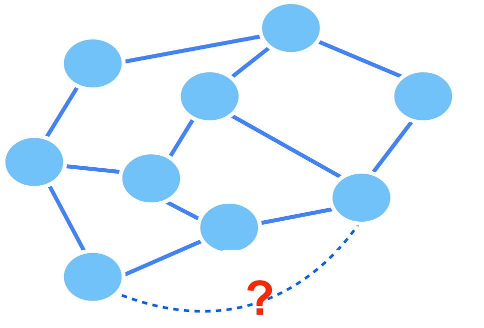
Construct a light robust similarity measure between not adjacent edges.



Problems

Similarity

Construct a light robust similarity measure between not adjacent edges.



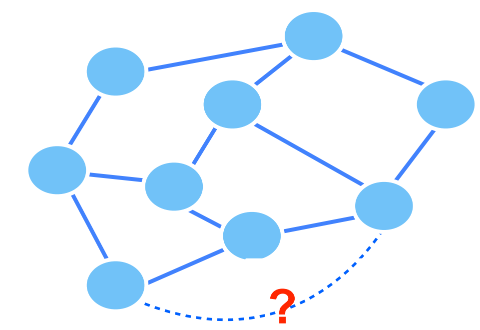
Motivation

Goal is to find list of similar nodes.

Problems

Similarity

Construct a light robust similarity measure between not adjacent edges.



Motivation

Goal is to find list of similar nodes.

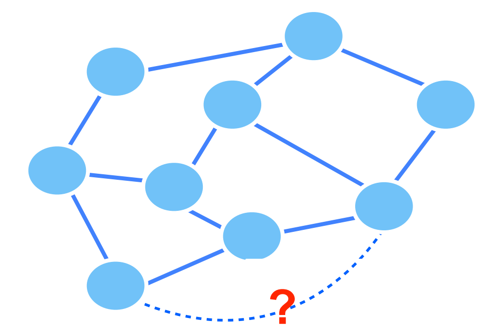
Connection with link prediction

Random walk based technique, number of paths, Jaccard similarity...

Problems

Similarity

Construct a light robust similarity measure between not adjacent edges.



Motivation

Goal is to find list of similar nodes.

Connection with link prediction

Random walk based technique, number of paths, Jaccard similarity...

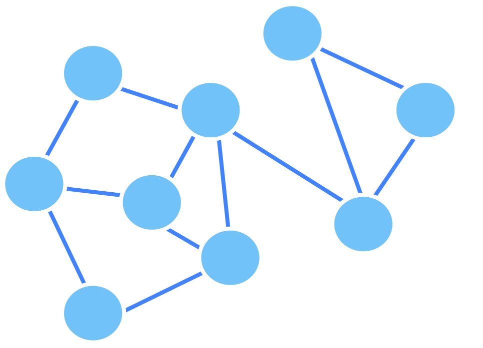
Several variations

Bipartite graphs, directed graphs...

Problems

Clustering

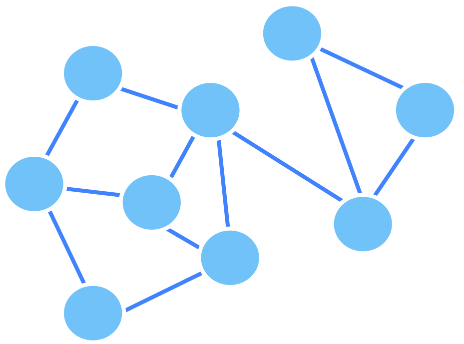
Find good clusters quickly in parallel.



Problems

Clustering

Find good clusters quickly in parallel.



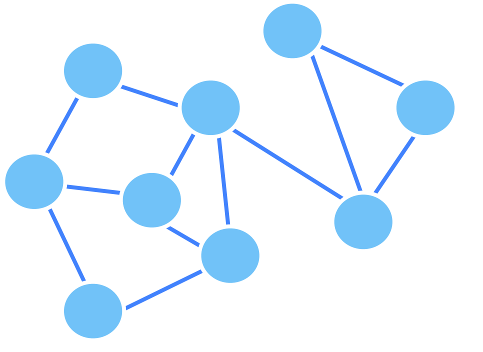
New challenges

Very large graphs, need of parallelizable solutions

Problems

Clustering

Find good clusters quickly in parallel.



New challenges

Very large graphs, need of parallelizable solutions

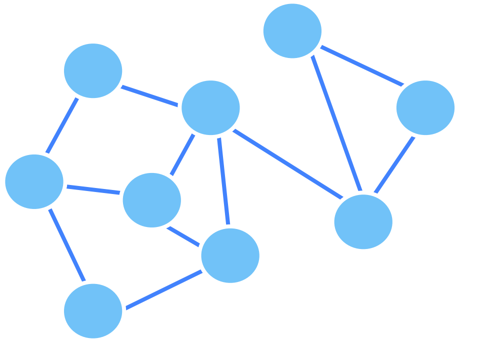
Few approaches

Random walks, hierarchical clustering, agglomerative clustering...

Problems

Clustering

Find good clusters quickly in parallel.



New challenges

Very large graphs, need of parallelizable solutions

Few approaches

Random walks, hierarchical clustering, agglomerative clustering...

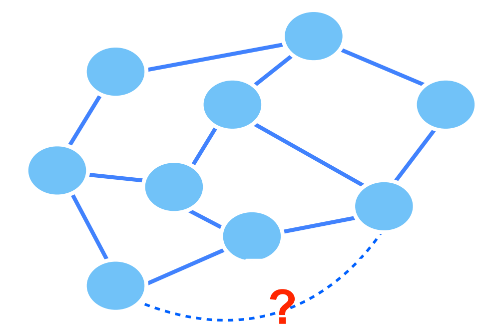
Different constraints

Balanced clustering, size constraint clustering...

A useful technique

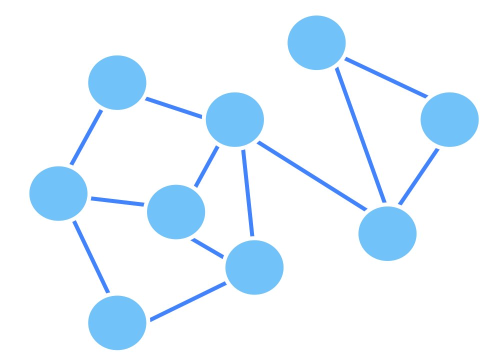
Similarity

Construct a light robust similarity measure between not adjacent edges.



Clustering

Find good clusters quickly in parallel.



Common approach based on random walk to solve both problems.

Local random walk and node similarities

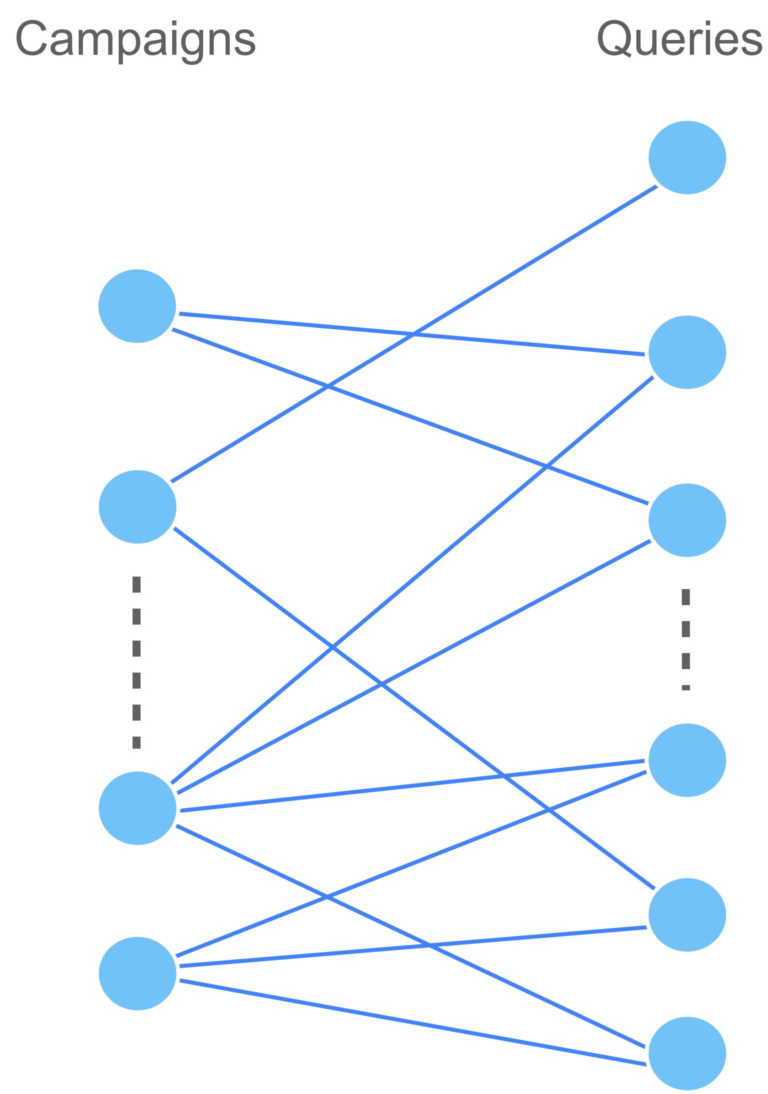
Joint work with:
Alessandro Epasto (Sapienza University)
Jon Feldman (Google Research NY)
Stefano Leonardi (Sapienza University)
Vahab Mirrokni (Google Research NY)
WWW 2014

A real world problem

Can we identify competitors of an Ads campaign?

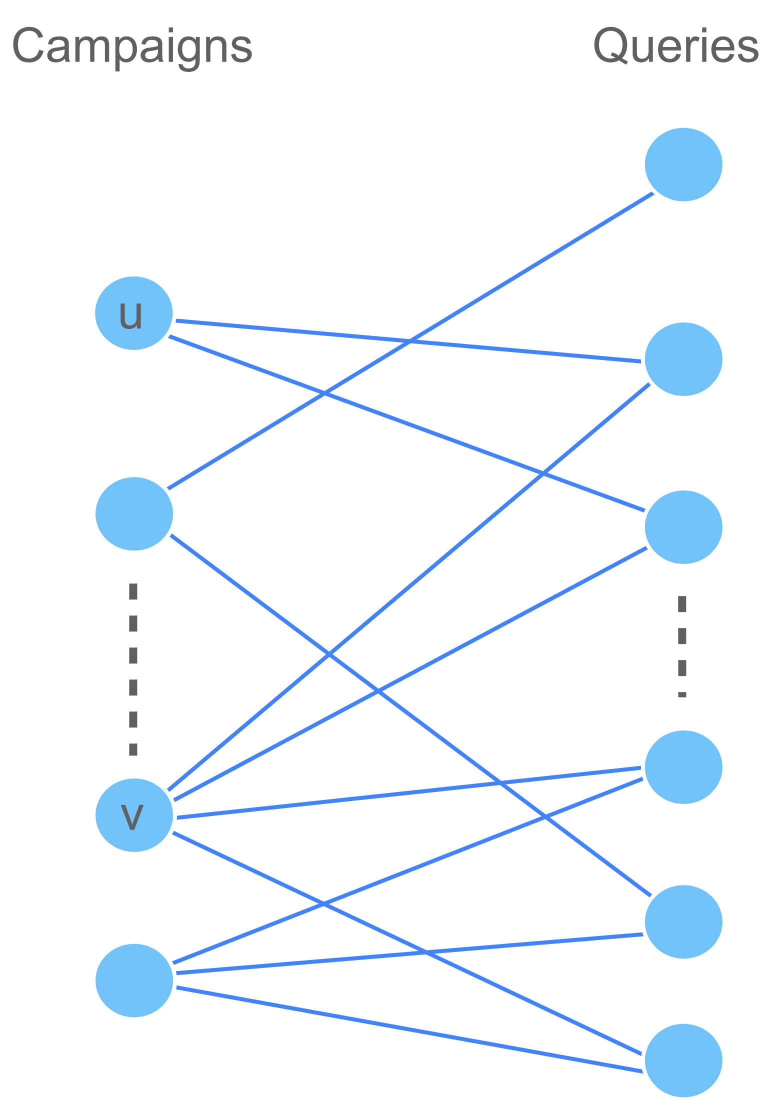
A real world problem

Can we identify competitors of an Ads campaign?



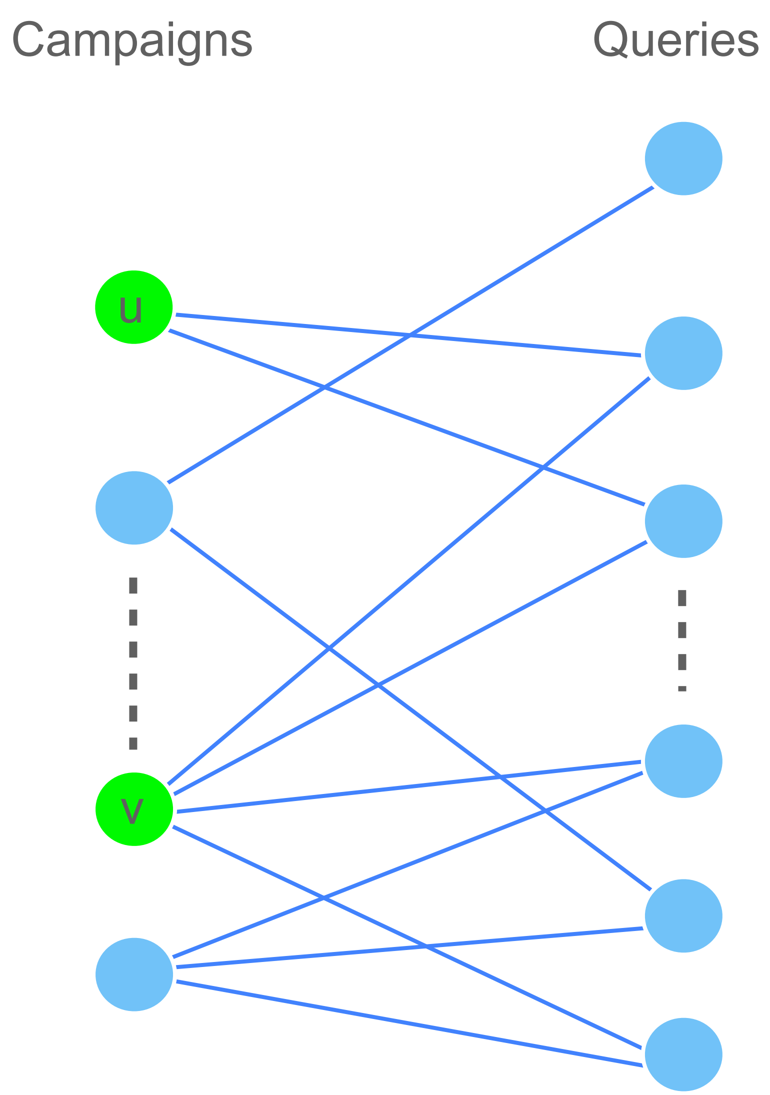
A real world problem

Various approaches



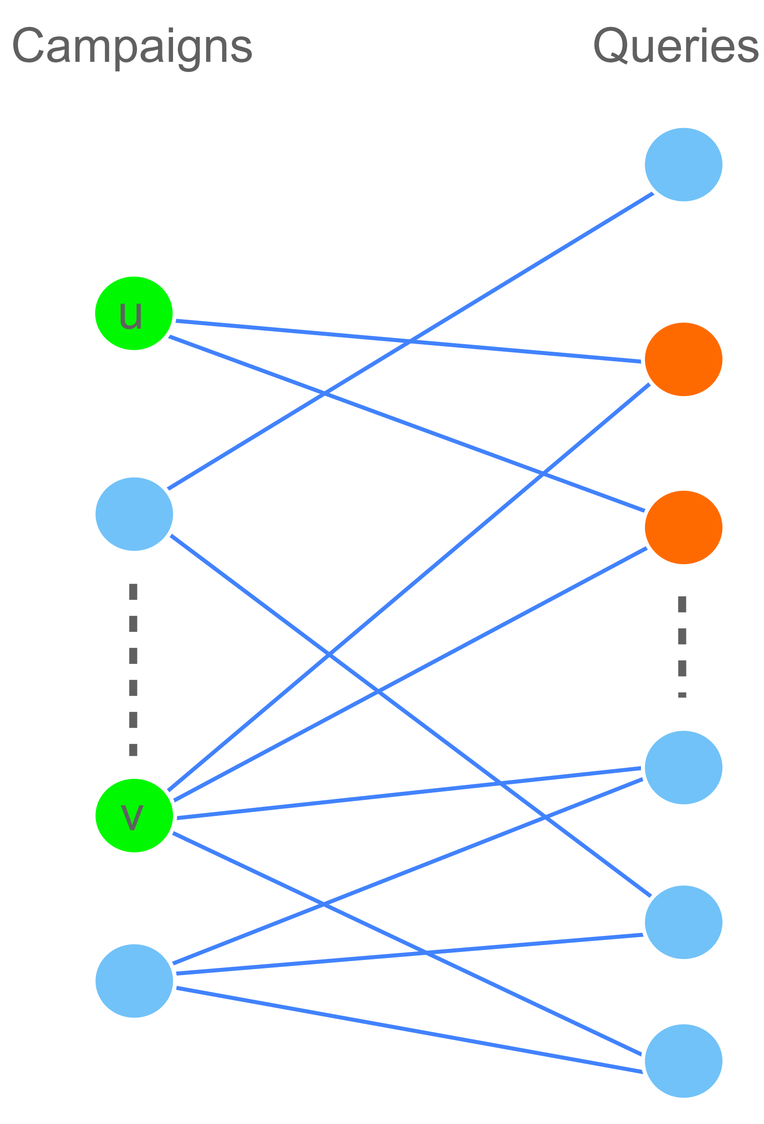
A real world problem

Various approaches



A real world problem

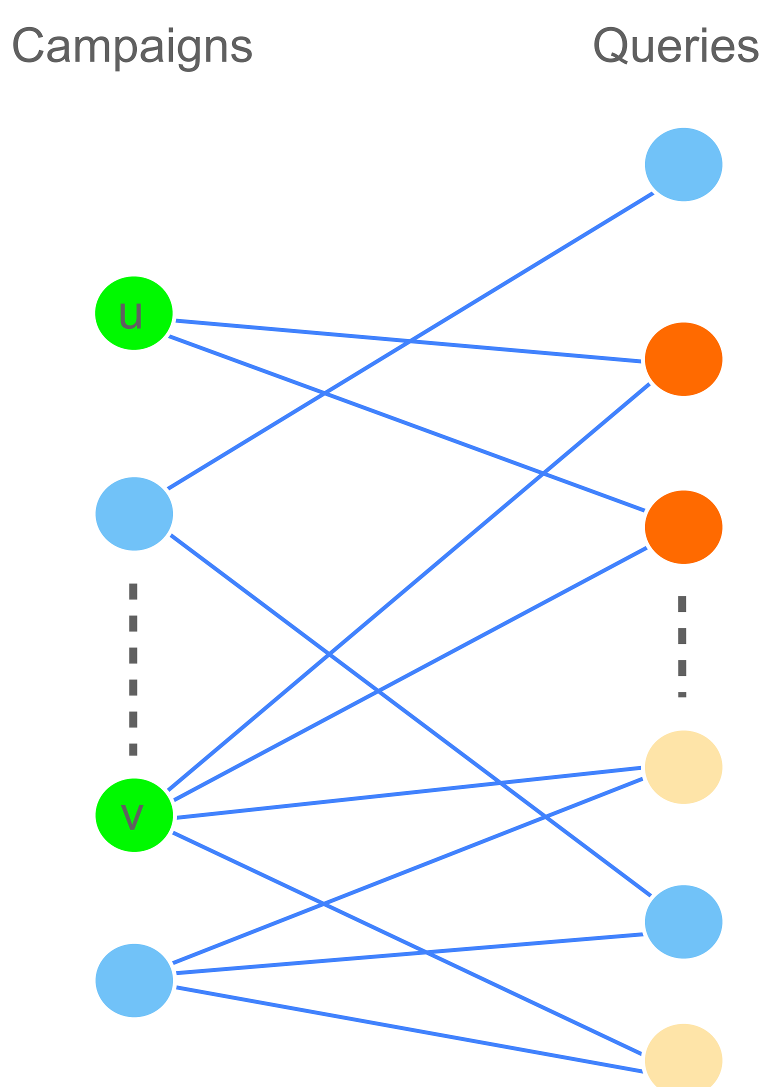
Various approaches



Common neighbors: 2

A real world problem

Various approaches

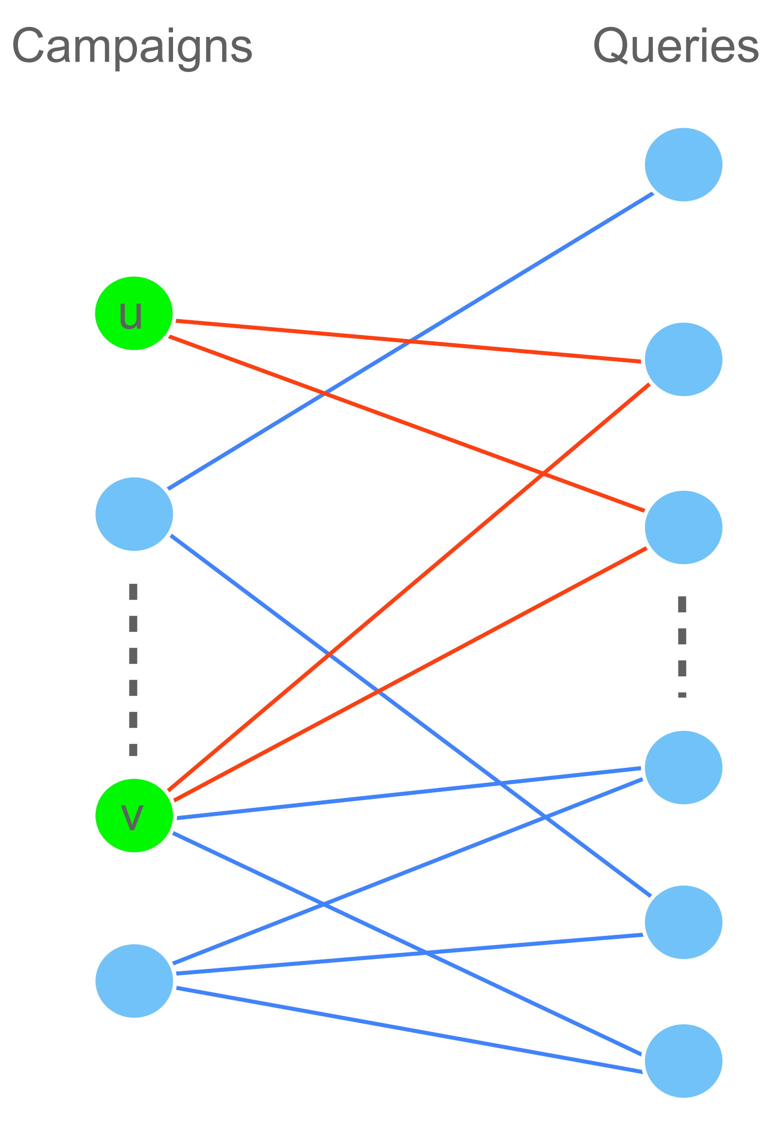


Common neighbors: 2

Jaccard similarity: $\frac{1}{2}$

A real world problem

Various approaches



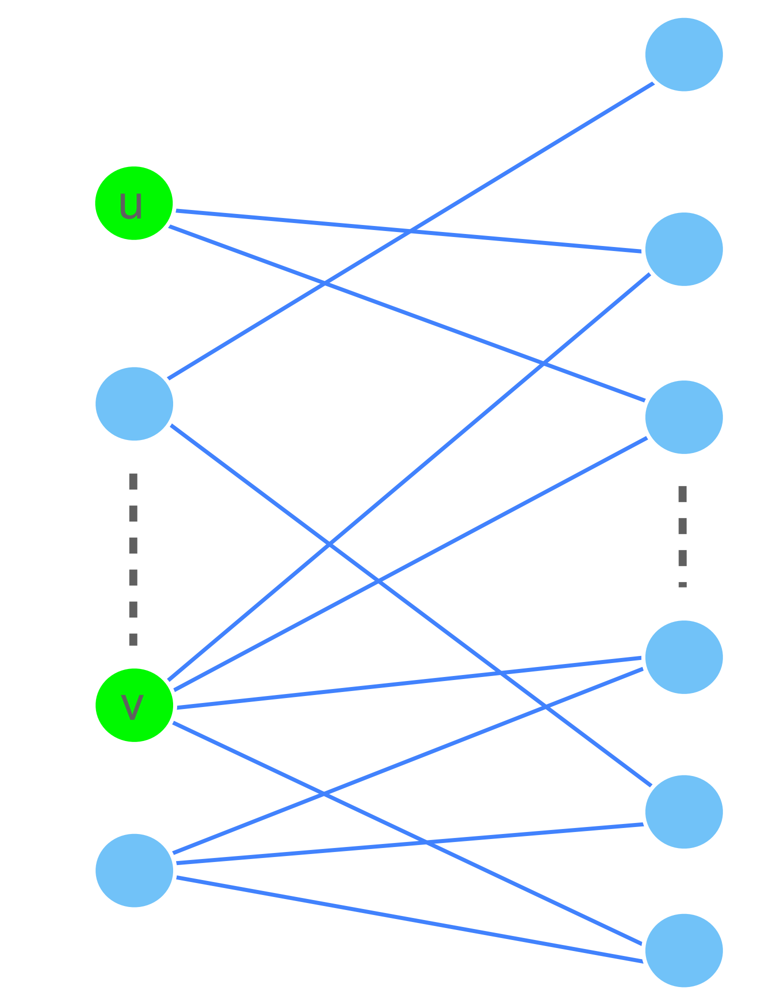
Common neighbors: 2

Jaccard similarity: $\frac{1}{2}$

Number of paths: 2

A real world problem

Various approaches



Common neighbors: 2

Jaccard similarity: $\frac{1}{2}$

Number of paths: 2

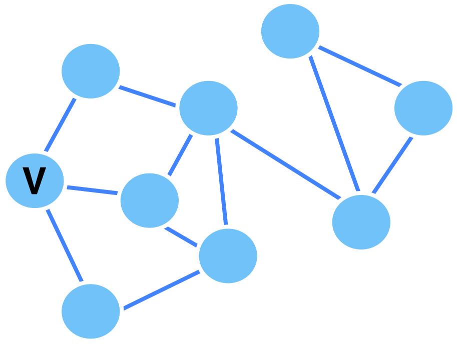
Short random walk(PPR)

Personalized PageRank

Personalized PageRank(v,u)

Probability of visiting *u* in the following lazy random walk: at each step,

- With probability $\frac{1}{2}\alpha$, go back to *v*.
- With probability $\frac{1}{2}(1-\alpha)$, go to a neighbor uniformly at random.

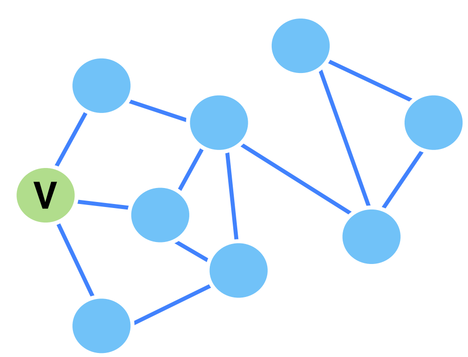


Personalized PageRank

Personalized PageRank(v,u)

Probability of visiting *u* in the following lazy random walk: at each step,

- With probability $\frac{1}{2}\alpha$, go back to *v*.
- With probability $\frac{1}{2}(1-\alpha)$, go to a neighbor uniformly at random.

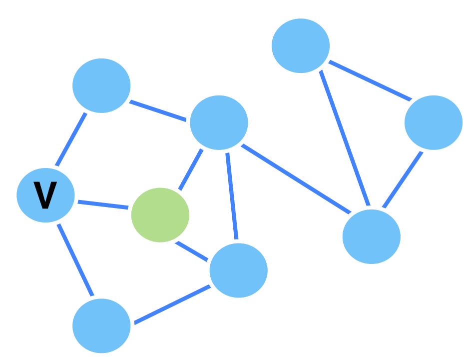


Personalized PageRank

Personalized PageRank(v,u)

Probability of visiting *u* in the following lazy random walk: at each step,

- With probability $\frac{1}{2}\alpha$, go back to *v*.
- With probability $\frac{1}{2}(1-\alpha)$, go to a neighbor uniformly at random.

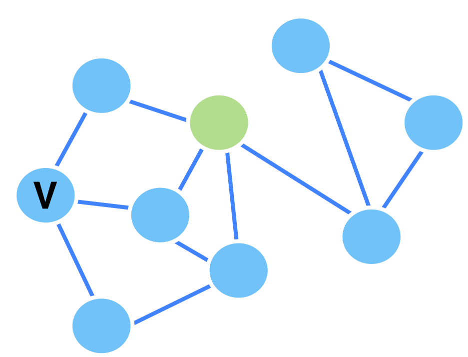


Personalized PageRank

Personalized PageRank(v,u)

Probability of visiting *u* in the following lazy random walk: at each step,

- With probability $\frac{1}{2}\alpha$, go back to *v*.
- With probability $\frac{1}{2}(1-\alpha)$, go to a neighbor uniformly at random.

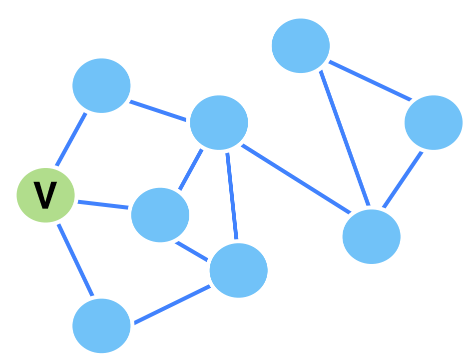


Personalized PageRank

Personalized PageRank(v,u)

Probability of visiting *u* in the following lazy random walk: at each step,

- With probability $\frac{1}{2}\alpha$, go back to *v*.
- With probability $\frac{1}{2}(1-\alpha)$, go to a neighbor uniformly at random.

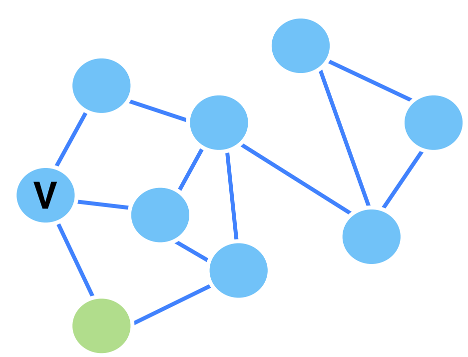


Personalized PageRank

Personalized PageRank(v,u)

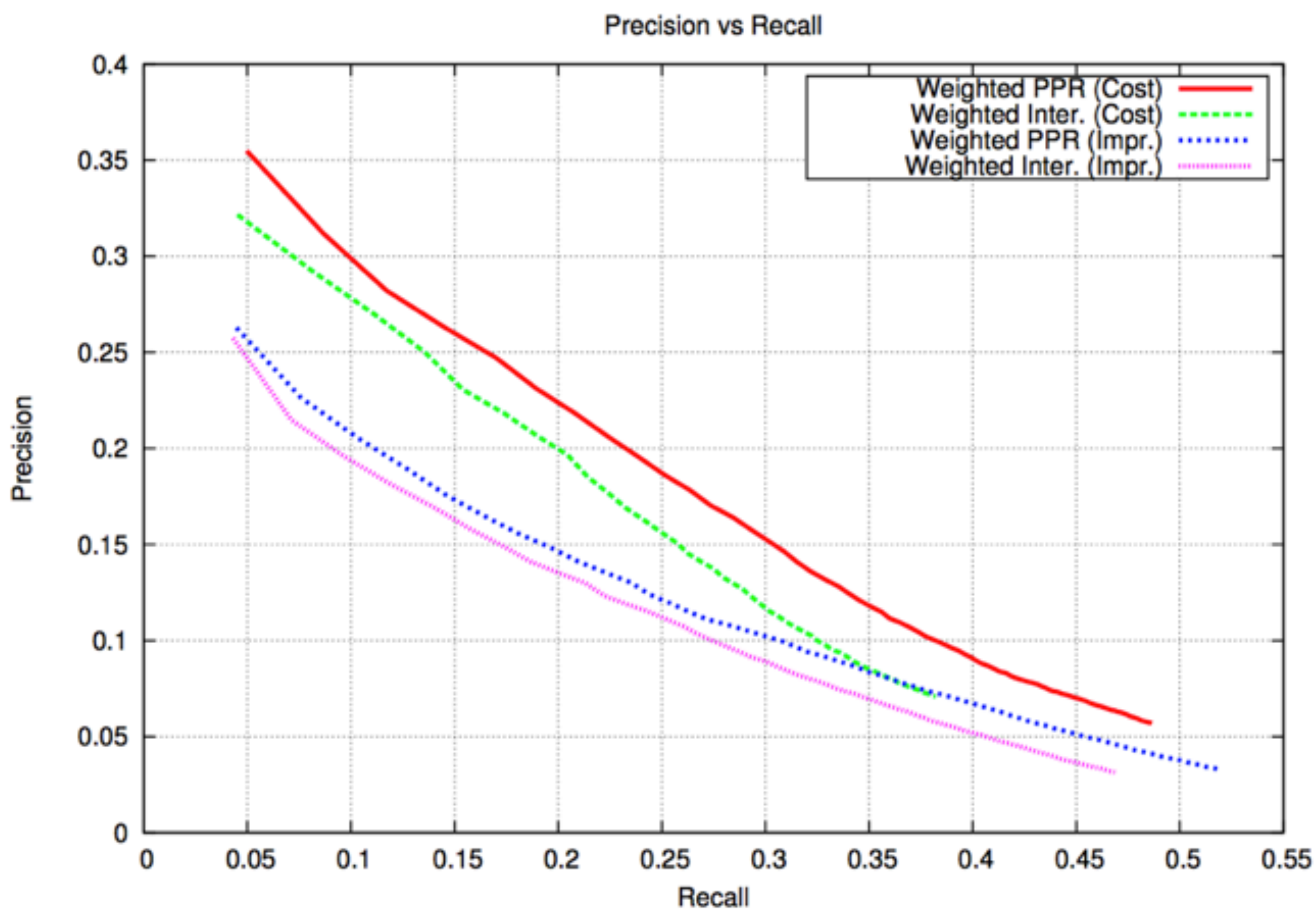
Probability of visiting *u* in the following lazy random walk: at each step,

- With probability $\frac{1}{2}\alpha$, go back to *v*.
- With probability $\frac{1}{2}(1-\alpha)$, go to a neighbor uniformly at random.



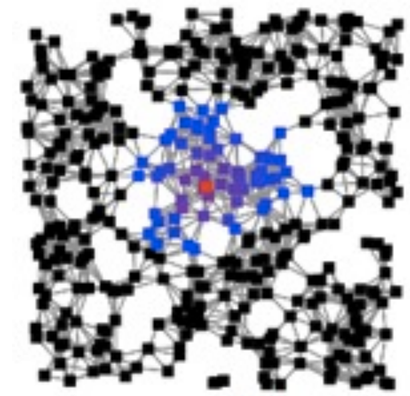
Experimental comparison

We had ground truth data

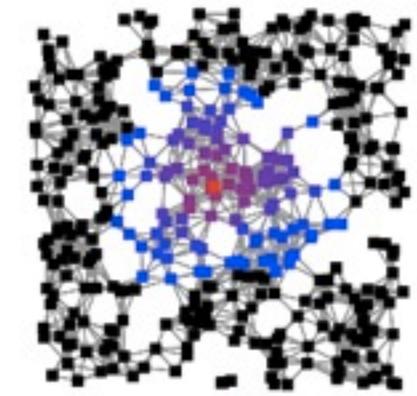


Approximate personalized PageRank

Approximate efficiently



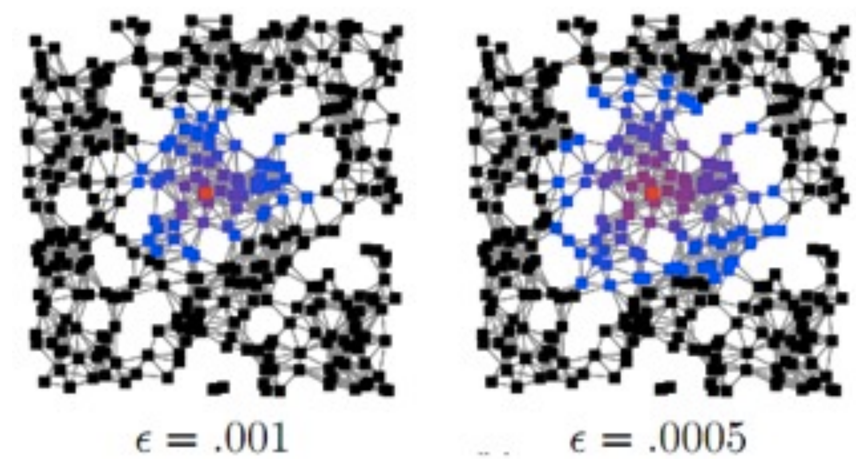
$\epsilon = .001$



$\epsilon = .0005$

Approximate personalized PageRank

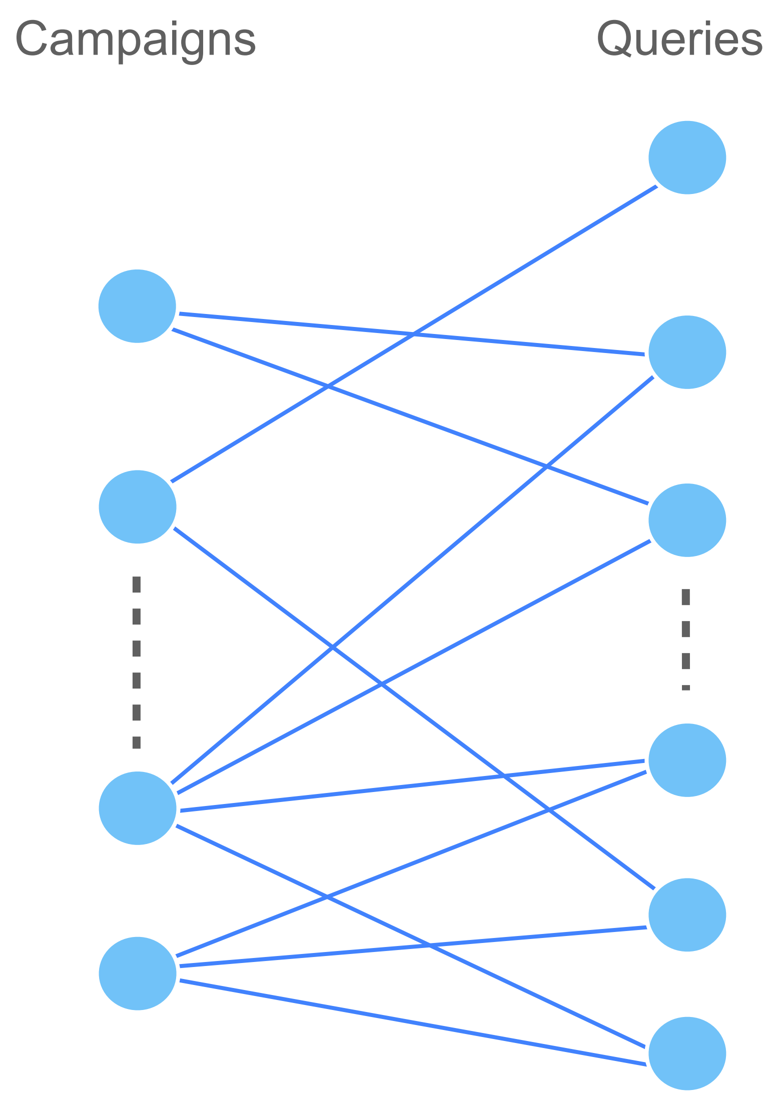
Approximate efficiently



Two main approach:
- Monte Carlo techniques
- Push-score techniques

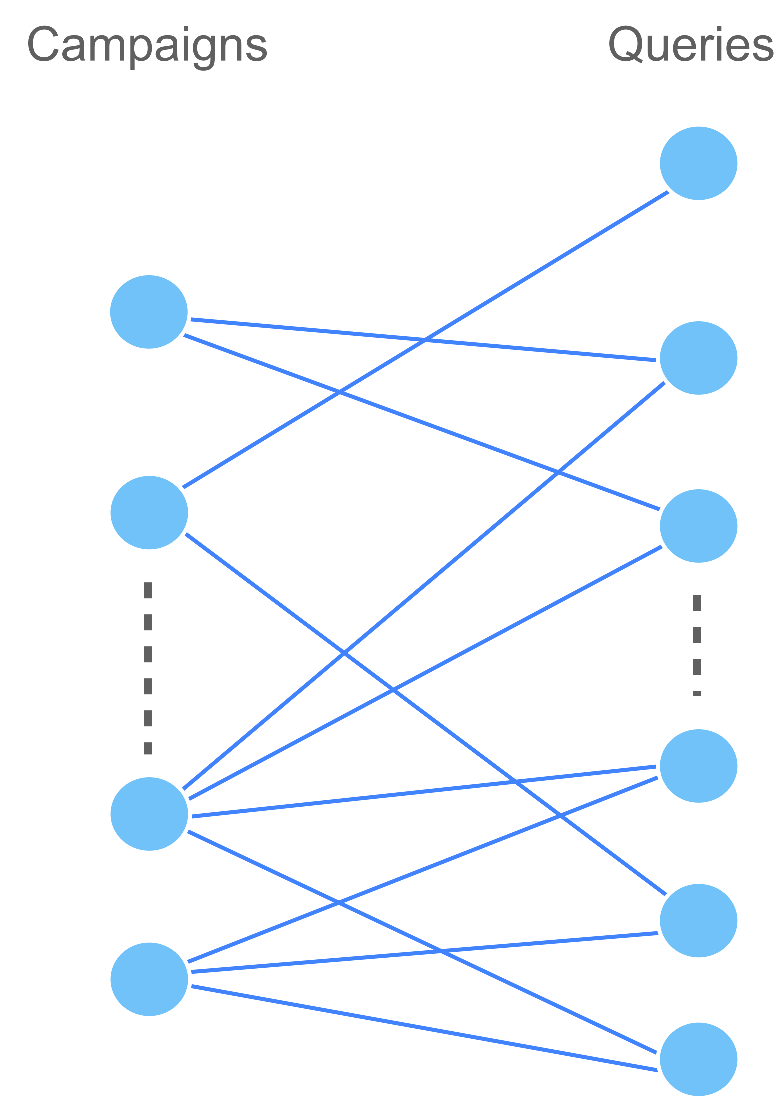
Large scale computations

The campaigns-queries graph is lopsided



Large scale computations

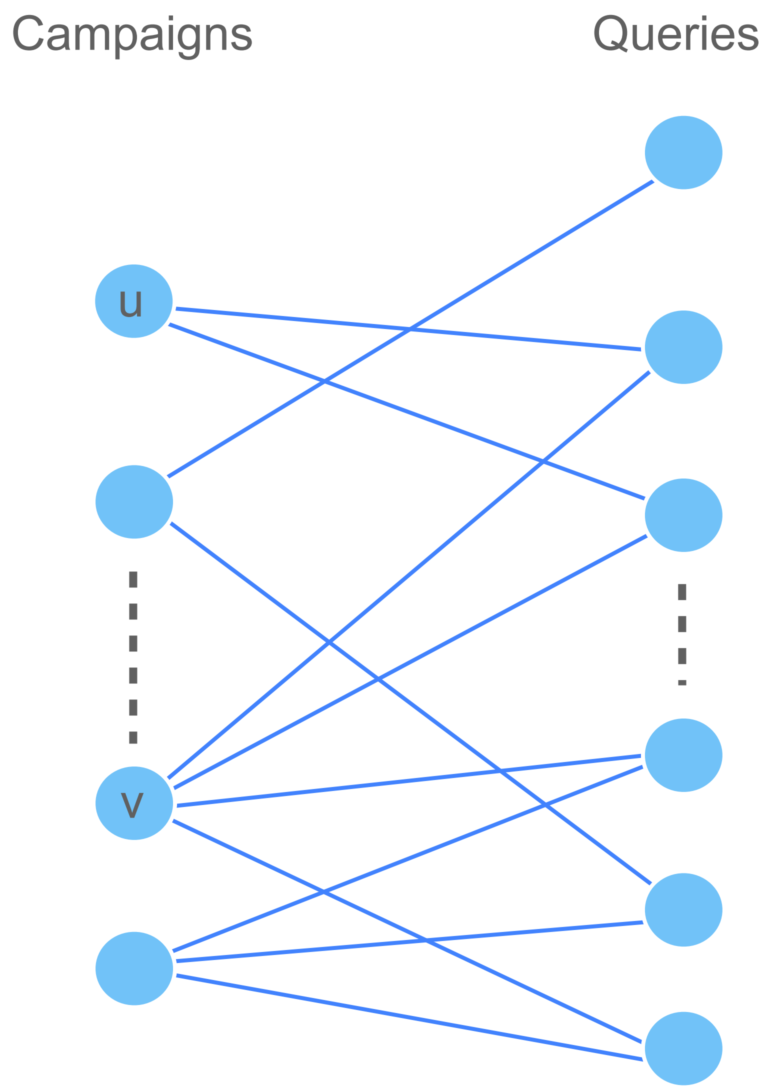
The campaigns-queries graph is lopsided



Millions of campaigns and hundreds of millions of queries.

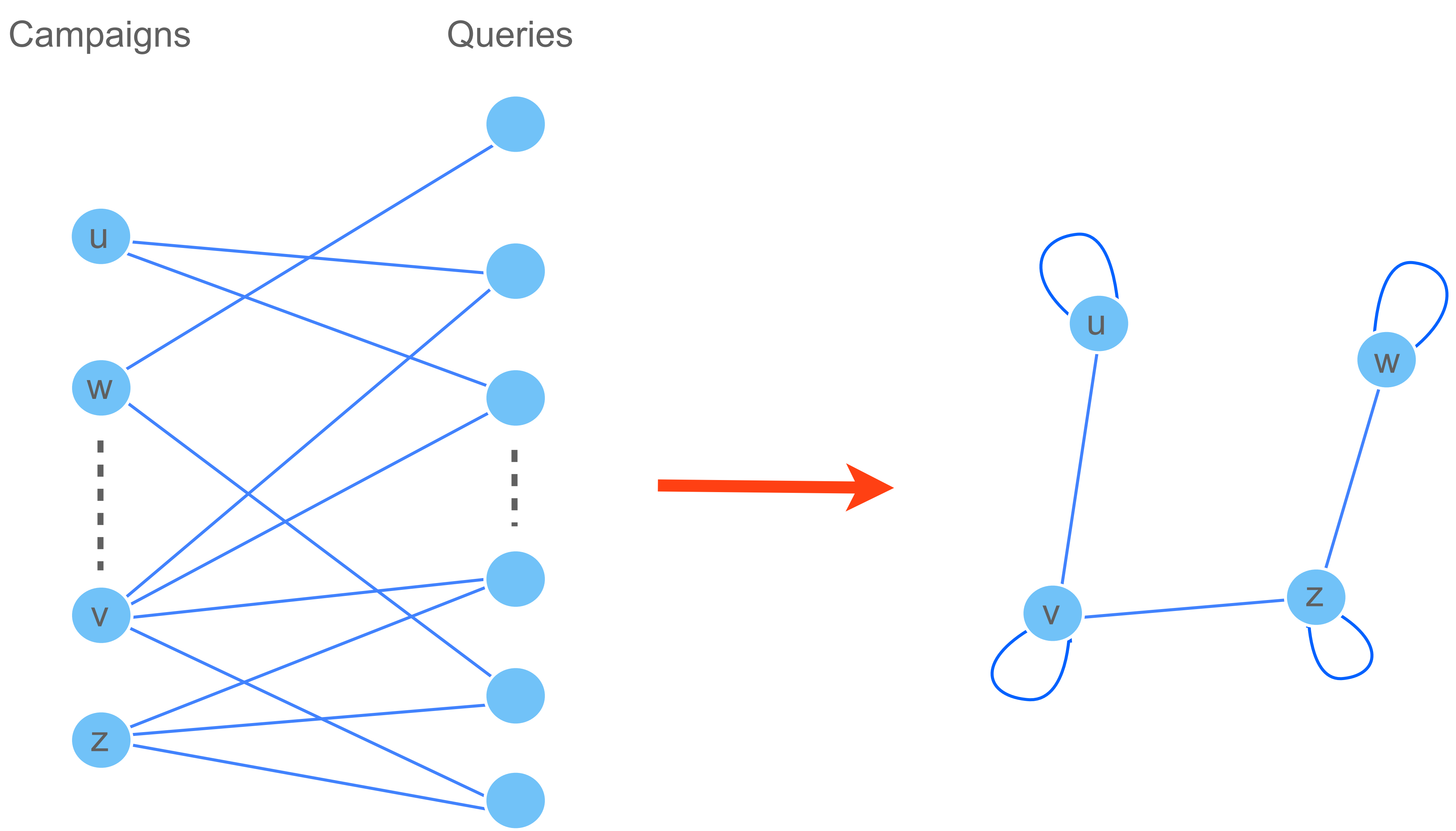
Large scale computations

We can reduce to a computation only on campaigns



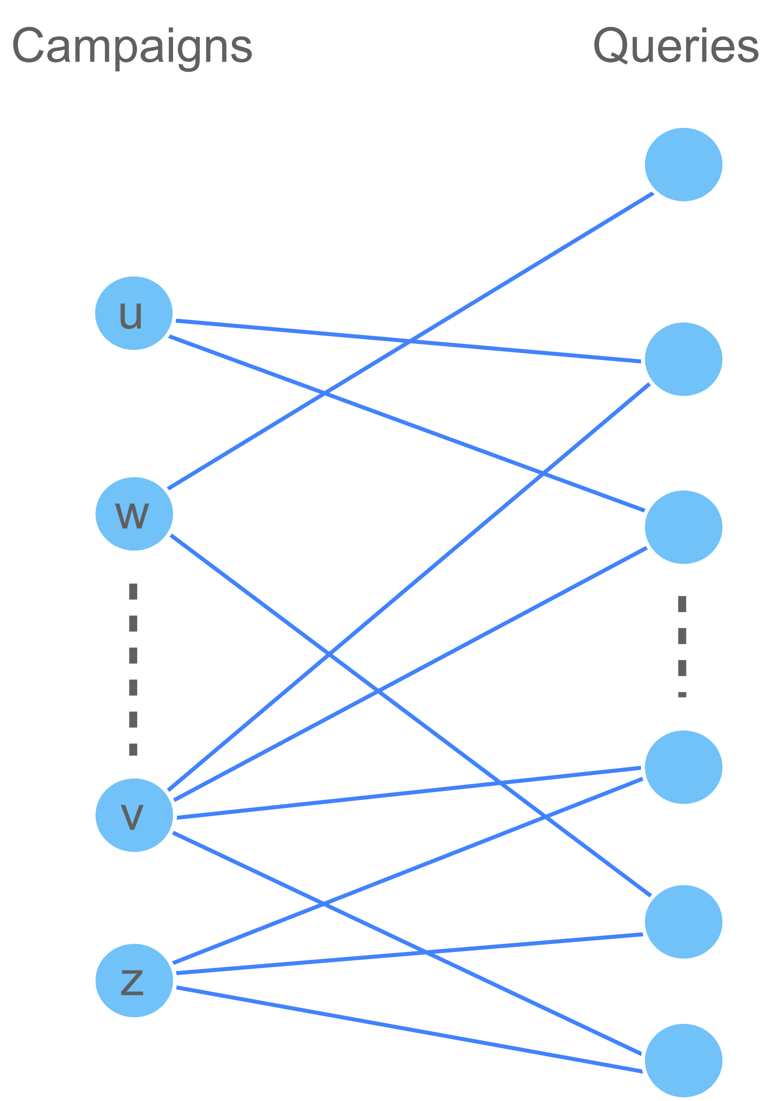
Large scale computations

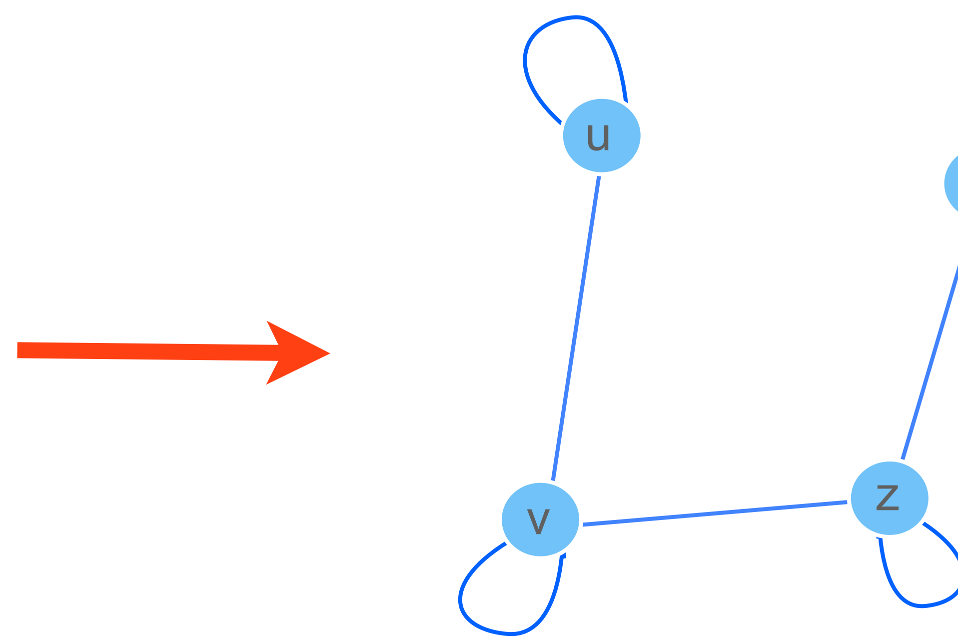
We can reduce to a computation only on campaigns



Large scale computations

We can reduce to a computation only on campaigns

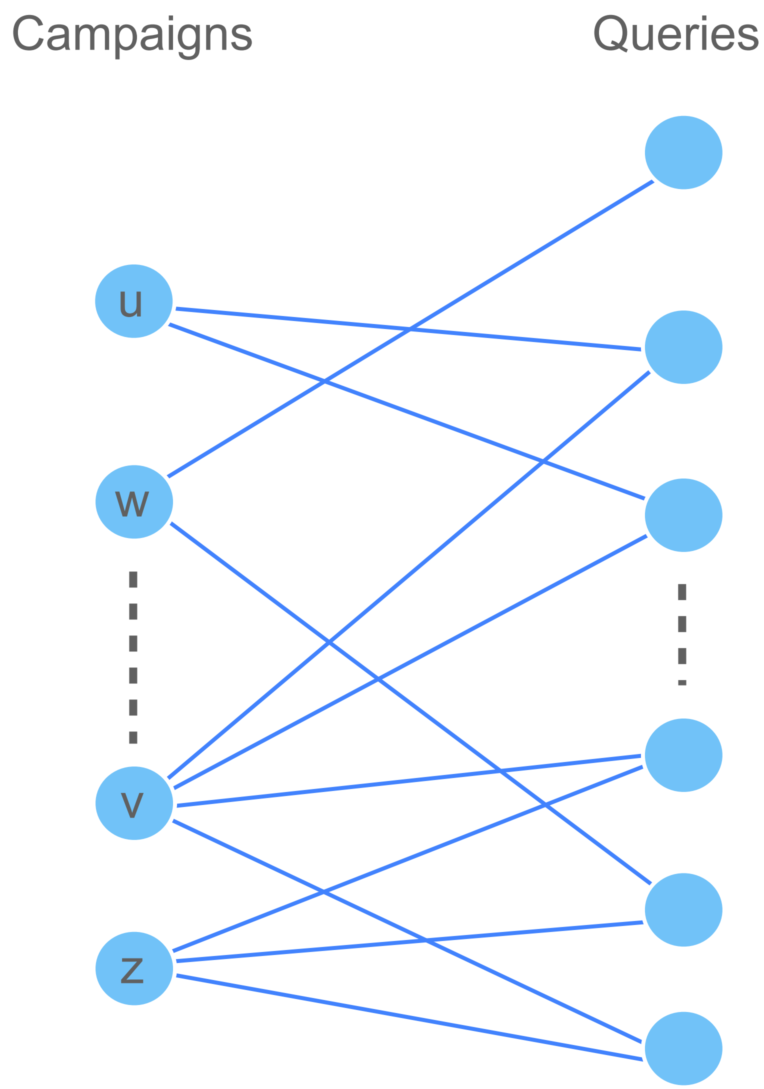


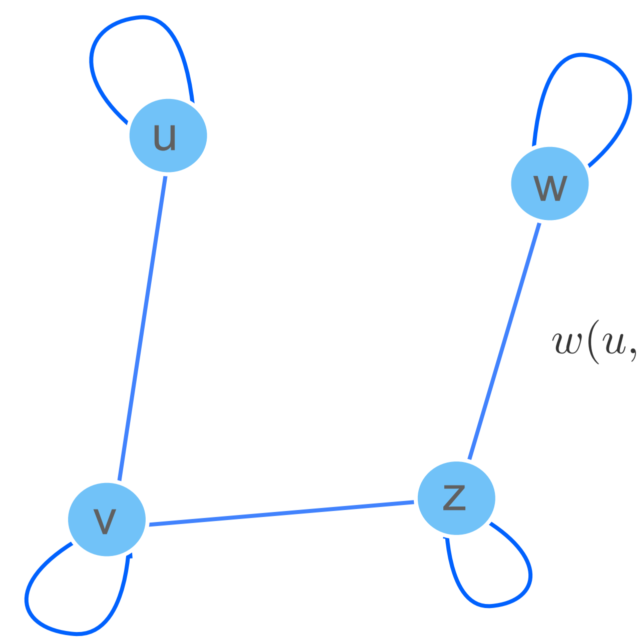


$$w(u,v) = \sum_{q \in N(u) \cap N(v)} \frac{w(u,q)w(q,v)}{d(q)}$$

Large scale computations

We can reduce to a computation only on campaigns



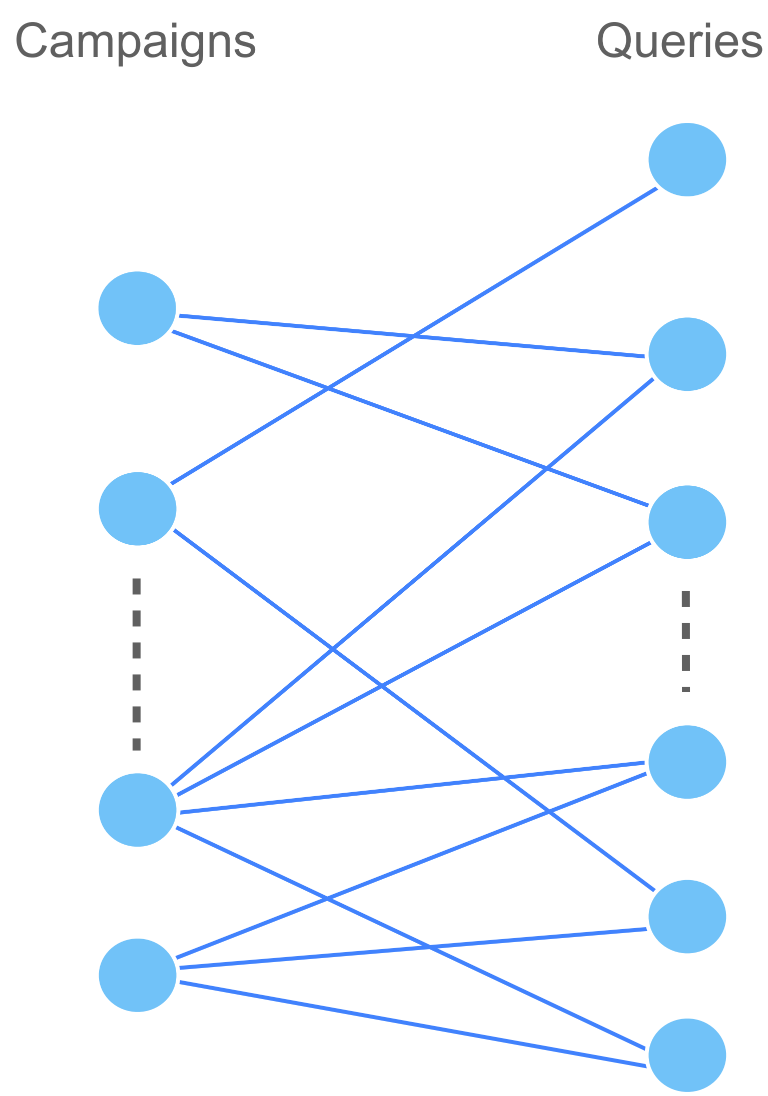


$$w(u,v) = \sum_{q \in N(u) \cap N(v)} \frac{w(u,q)w(q,v)}{d(q)}$$

$$PPR(u,v)_B = \frac{1}{2-\alpha} PPR(u,v)_G$$

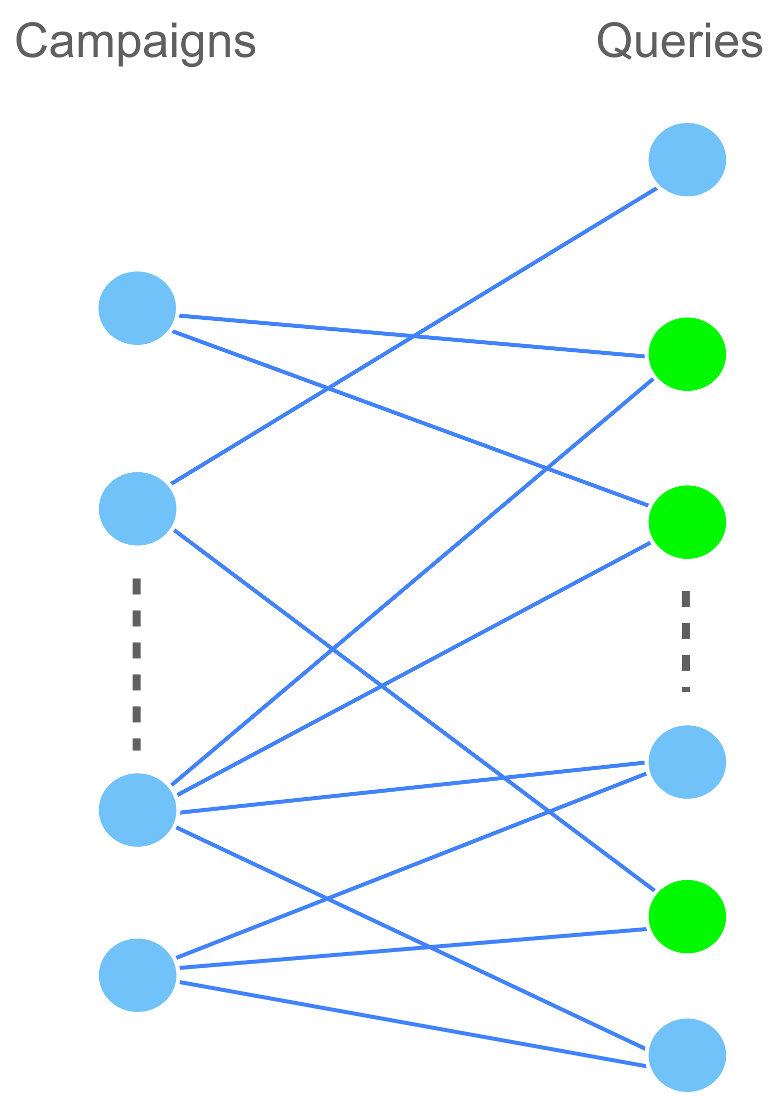
Extensions

Can we identify competitors of an Ads campaign in a specific category?



Extensions

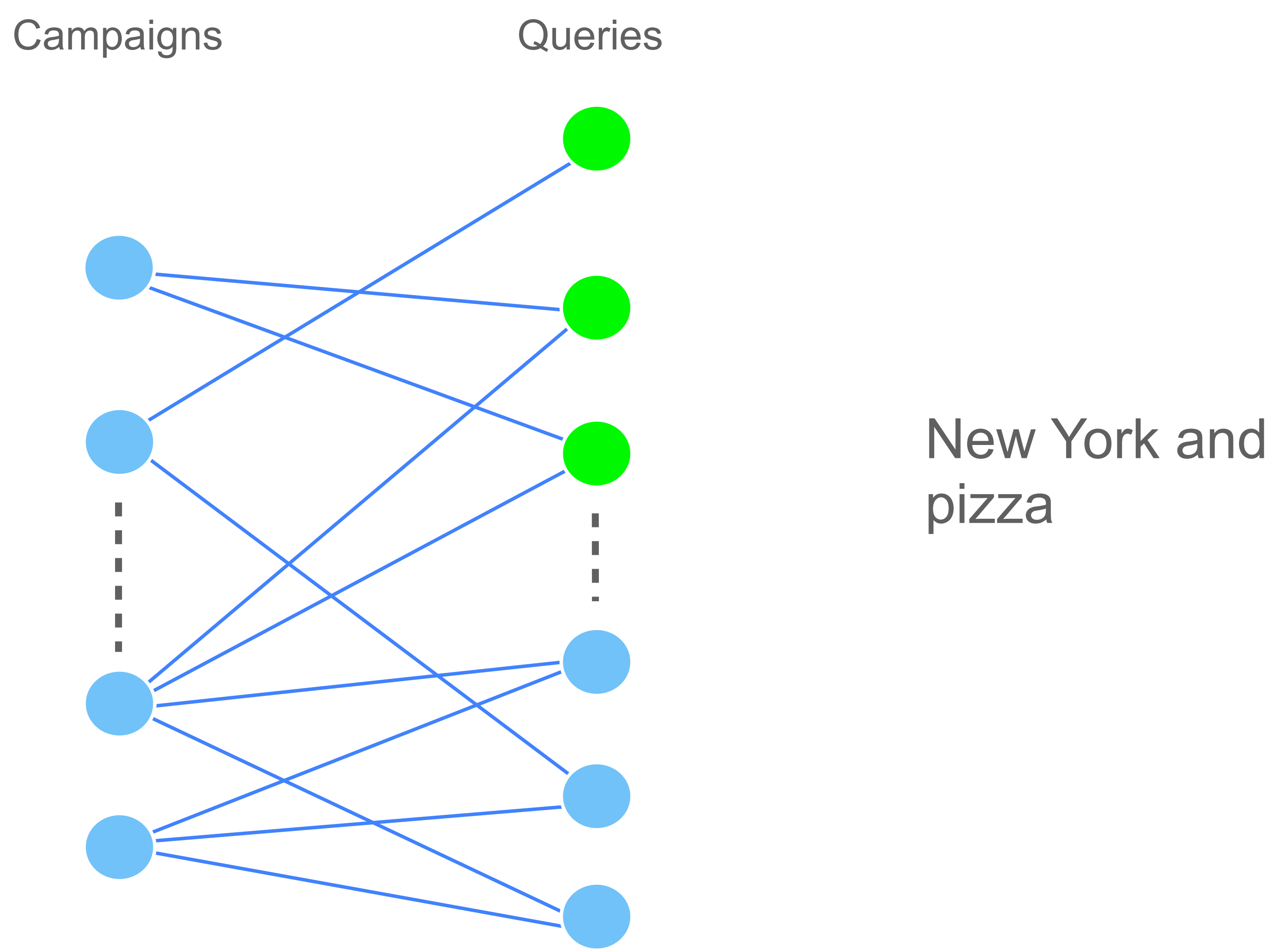
Can we identify competitors of an Ads campaign in a specific category?



New York and sport

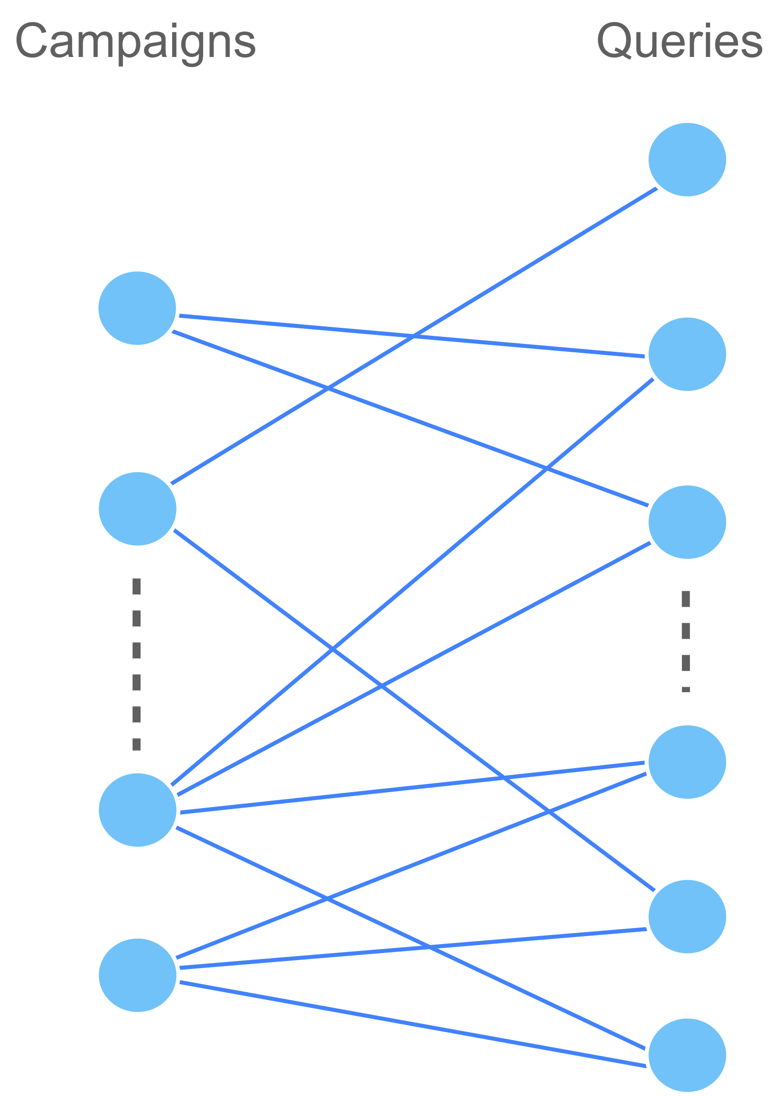
Extensions

Can we identify competitors of an Ads campaign in a specific category?



Extension

Can we identify competitors of an Ads campaign in a specific category?



Also in this setting by using some pre-computation we can compute the PPR efficiently.

Local random walk and clustering in practice

Joint work with:
Raimondas Kiveris (Google Research NY)
Vahab Mirrokni (Google Research NY)

Some basic intuitions

It would be nice to have the number and the length all the possible paths between two nodes.

Some basic intuitions

It would be nice to have the number and the length all the possible paths between two nodes.

Infeasible.

Some basic intuitions

It would be nice to have the number and the length all the possible paths between two nodes.

Infeasible.

We are interested just in strong relationship, we can sample.

Truncated random walk techniques

Run several truncated random walk of a specific length.

Truncated random walk techniques

Run several truncated random walk of a specific length.

Local algorithms based on this intuition:

- Truncated random walk, Personalized PageRank, Evolving set

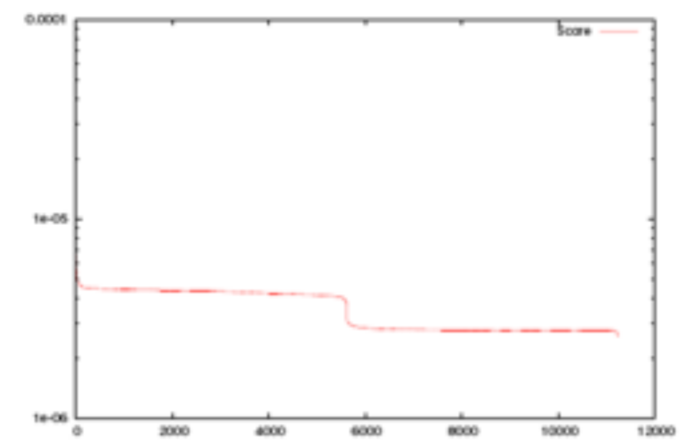
Nice experimental properties of PPR

We can approximate it efficiently in MapReduce by analyzing short random walks recursively.

Nice experimental properties of PPR

We can approximate it efficiently in MapReduce by analyzing short random walks recursively.

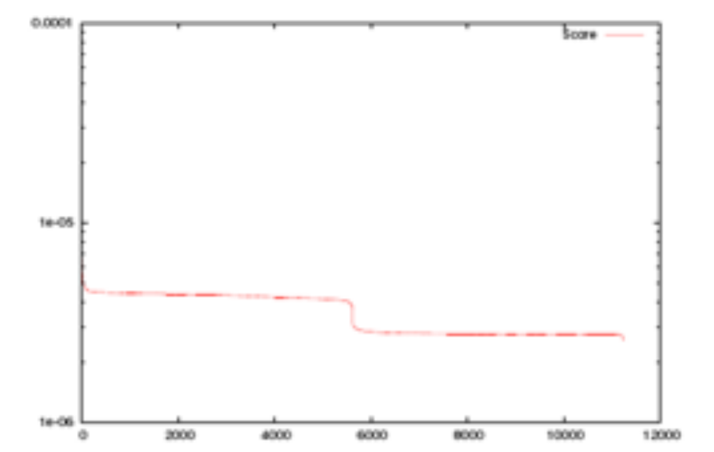
It works well in synthetic settings



Nice experimental properties of PPR

We can approximate it efficiently in MapReduce by analyzing short random walks recursively.

It works well in synthetic settings

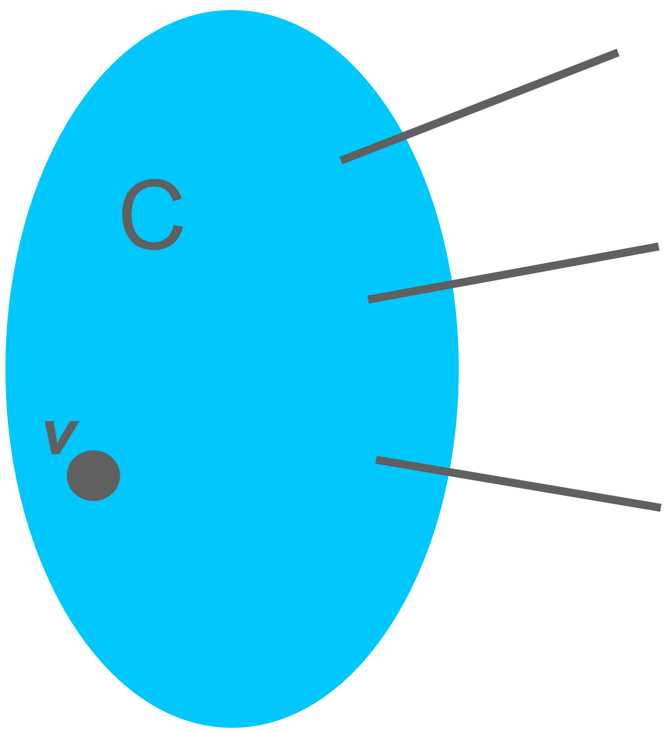


It works well in practice:

* On public graphs with 8M nodes
  -- Overlapping Clustering and Distributed Computation (WSDM'11, Andersen, Gleich, Mirrokni)
* On YouTube co-watch Graph with 100M nodes with 100s of machines
  -- Large-scale Community Detection on Youtube graph (ICWSM'11, Gargi, Lu, Mirrokni, Yoon)
* For sybil detection in social networks
  -- The evolution of Sybil Defense via Social Networks (S&P'13, Alvisi, Clement, Epasto, Lattanzi, Panconesi)

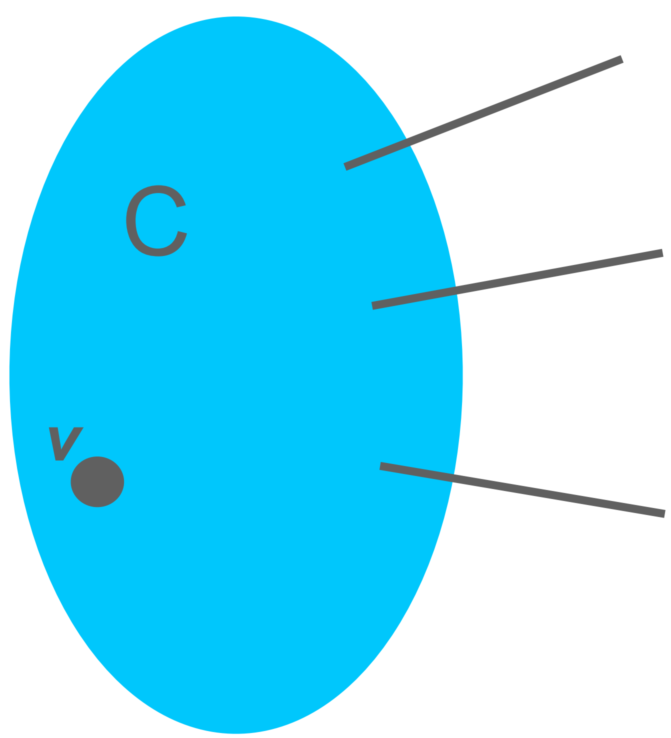
Why does it work?

Suppose to have a set with few edges going outside



Why does it work?

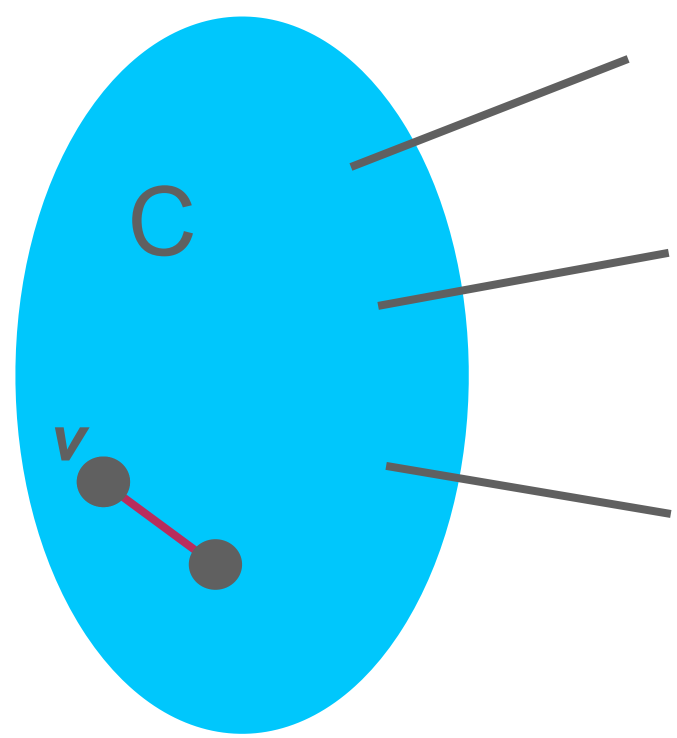
Suppose to have a set with few edges going outside



Most of the time a random walk will stay in C

Why does it work?

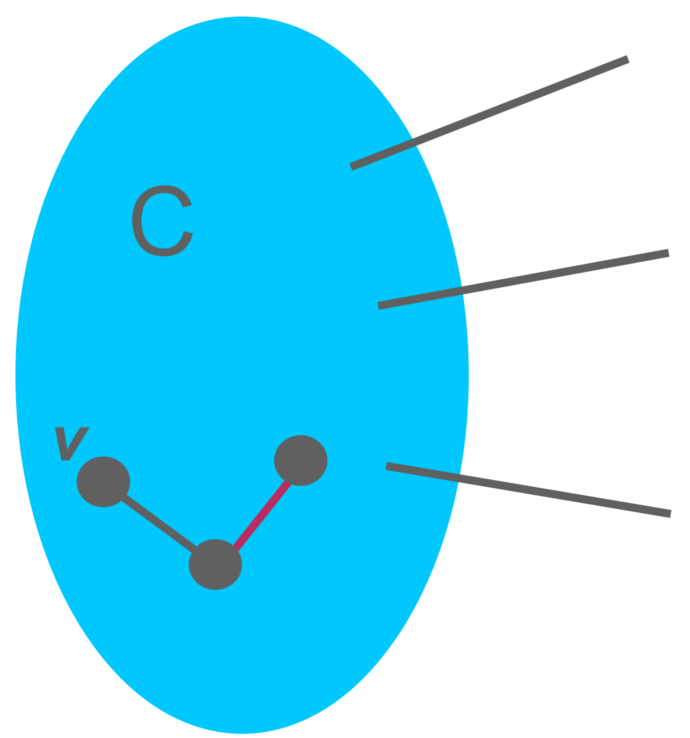
Suppose to have a set with few edges going outside



Most of the time a random walk will stay in C

Why does it work?

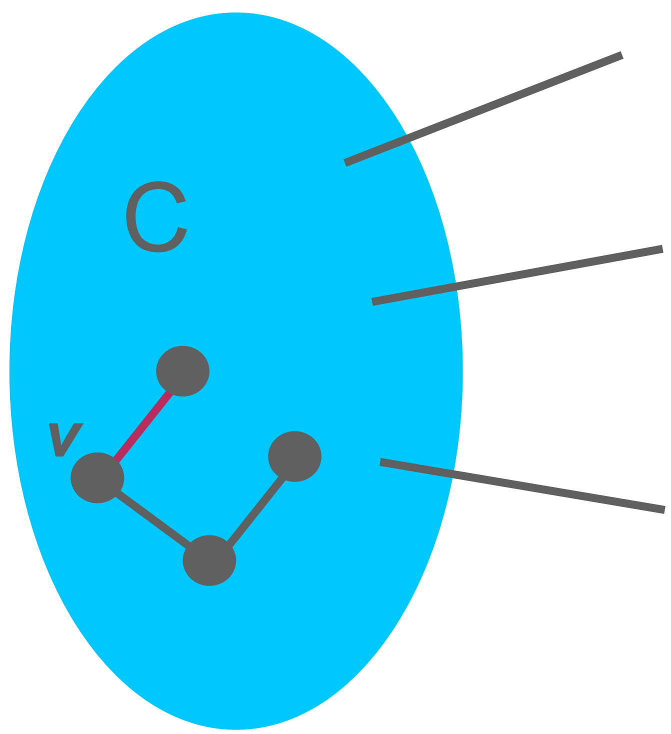
Suppose to have a set with few edges going outside



Most of the time a random walk will stay in C

Why does it work?

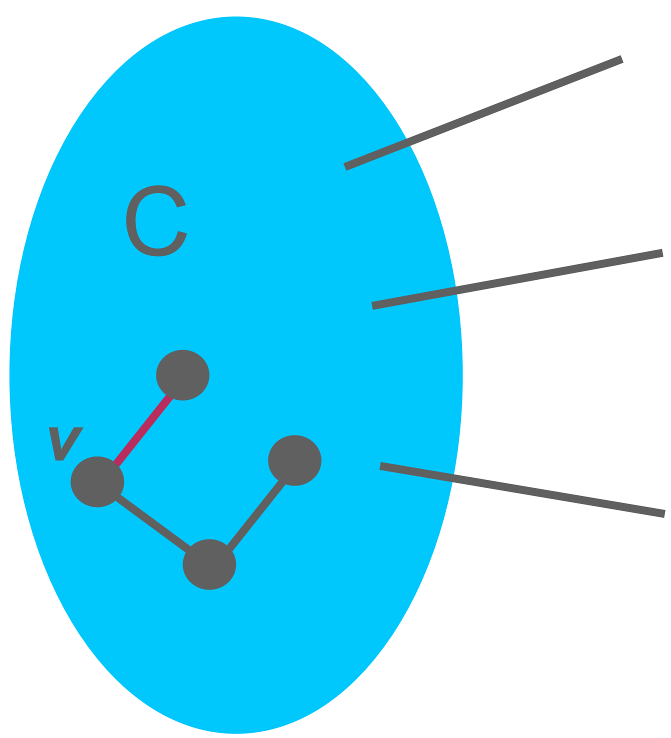
Suppose to have a set with few edges going outside



Most of the time a random walk will stay in C

Why does it work?

Suppose to have a set with few edges going outside



Most of the time a random walk will stay in C

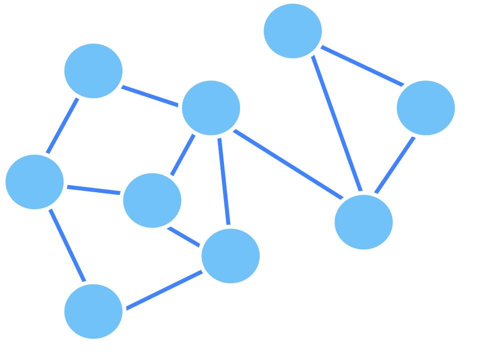
It is possible to bound the amount of score that goes outside C

Local clustering via random walk

How should we define a cluster?

Good clusters have cut conductance ϕ

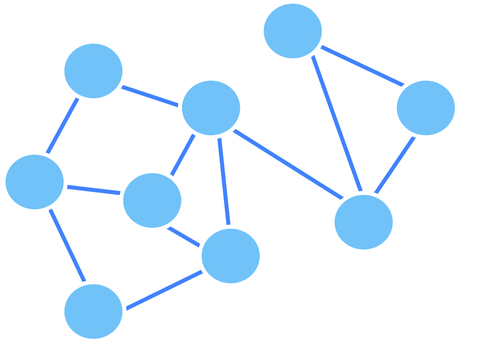
$$\phi = \frac{|cut(C, V-C)|}{min(Vol(C), Vol(V-C))}$$

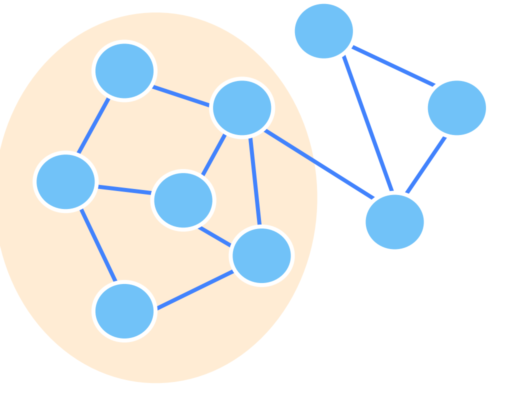


How should we define a cluster?

Good clusters have cut conductance ϕ

$$\phi = \frac{|cut(C, V - C)|}{min(Vol(C), Vol(V - C))}$$





$$\frac{1}{17}$$

Set of minimum conductance

Problem is NP-hard

Algorithms:

$\phi(S) = O(\sqrt{\phi})$ Spectral algorithms [Jerrum&Sinclair'89]

$\phi(S) = O(\log n)\phi$ [Leighten-Rao'99]

$\phi(S) = O(\sqrt{\log n})\phi$ [Arora-Rao-Vazirani'04]

Set of minimum conductance

Problem is NP-hard

Algorithms:

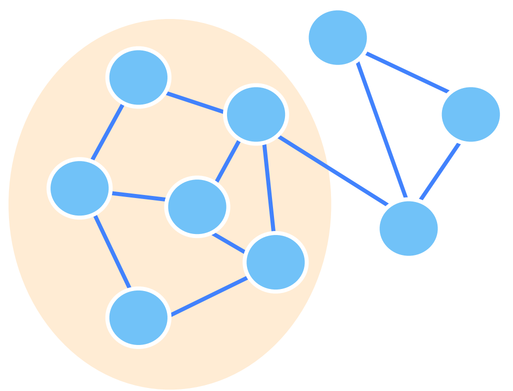
$\phi(S) = O(\sqrt{\phi})$ Spectral algorithms [Jerrum&Sinclair'89]

$\phi(S) = O(\log n)\phi$ [Leighten-Rao'99]

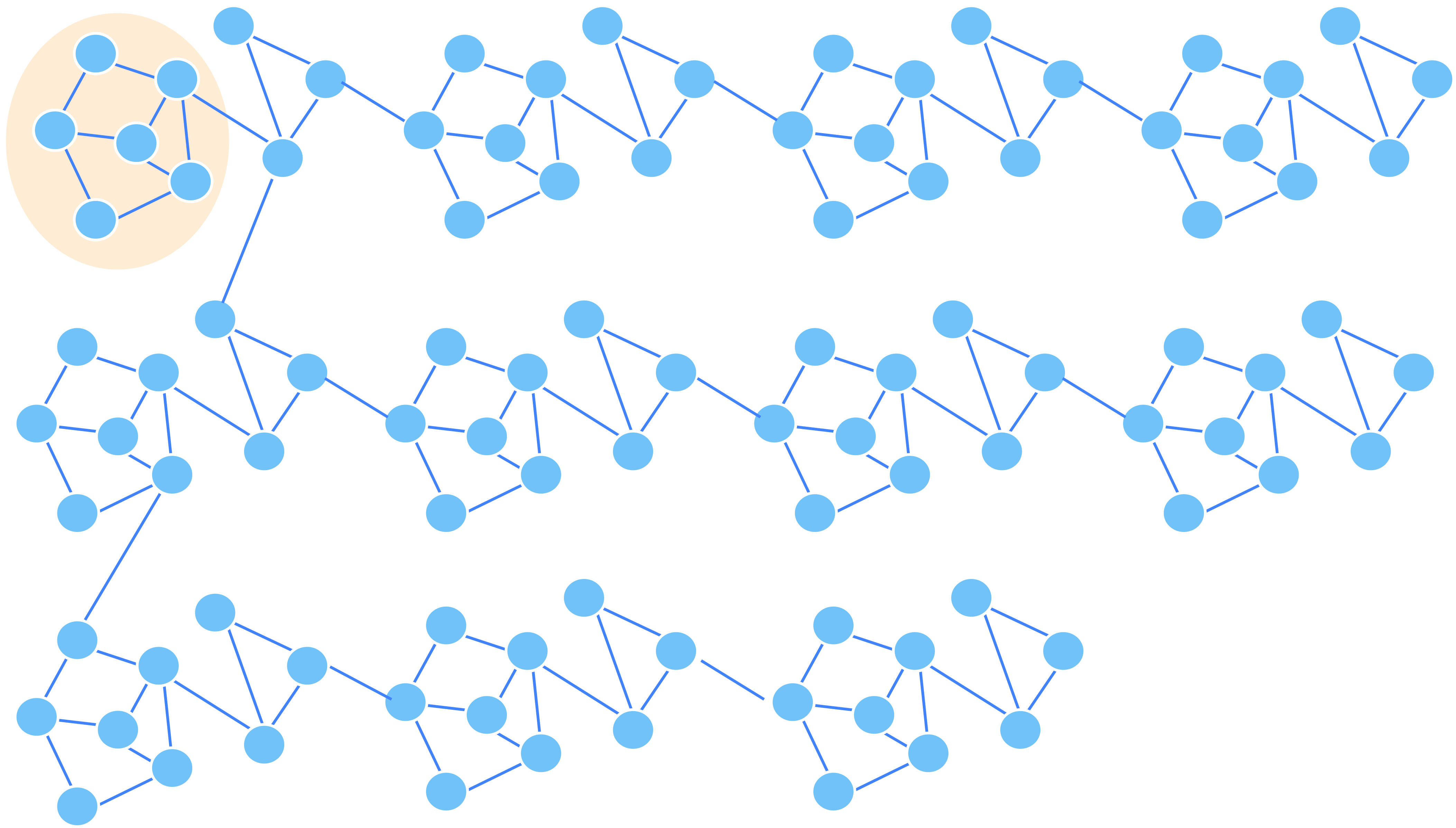
$\phi(S) = O(\sqrt{\log n})\phi$ [Arora-Rao-Vazirani'04]

Running time is at least linear in the size of the graph...

Local Graph Clustering



Local Graph Clustering

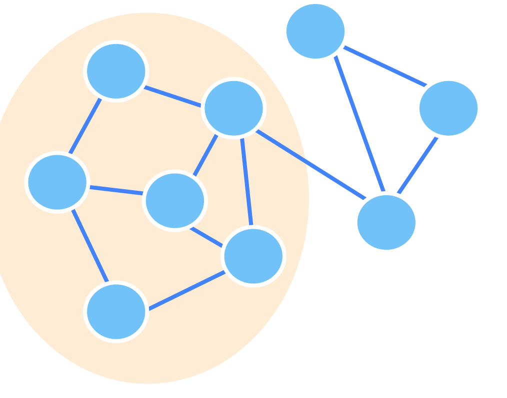


Do we really need to explore all the graph?!?

Local Clustering Algorithm

Given a good node v, the algorithm:

- Returns a set around v of good conductance
- Runs in time proportional to the size of the output
- Explores only the local neighborhood of v
- Returns a set with roughly the same size of S



Previous results

	Approximation guarantee	Running time
Truncated random walk [Spielman-Teng'04]	$\phi^{\frac{1}{3}} \log^{\frac{2}{3}} n$	$\tilde{O}\left(\frac{Vol(S)}{\phi^{5/3}}\right)$
Truncated random walk [Spielman-Teng'08]	$\sqrt{\phi} \log^{\frac{3}{2}} n$	$\tilde{O}\left(\frac{Vol(S)}{\phi^{2}}\right)$
PageRank random walk [Andersen-Chung-Lang'06]	$\sqrt{\phi \log n}$	$\tilde{O}\left(\frac{Vol(S)}{\phi}\right)$
Evolving Set [Andersen-Peres'08]	$\sqrt{\phi \log n}$	$\tilde{O}\left(\frac{Vol(S)}{\sqrt{\phi}}\right)$
Evolving Set [Gharan-Trevisan'12]	$\sqrt{\frac{\phi}{\epsilon}}$	$\tilde{O}\left(\frac{Vol(S)^{1+\epsilon}}{\sqrt{\phi}}\right)$

Previous results

	Approximation guarantee	Running time
Truncated random walk [Spielman-Teng'04]	$\phi^{\frac{1}{3}} \log^{\frac{2}{3}} n$	$\tilde{O}\left(\frac{Vol(S)}{\phi^{5/3}}\right)$
Truncated random walk [Spielman-Teng'08]	$\sqrt{\phi} \log^{\frac{3}{2}} n$	$\tilde{O}\left(\frac{Vol(S)}{\phi^{2}}\right)$
PageRank random walk [Andersen-Chung-Lang'06]	$\sqrt{\phi \log n}$	$\tilde{O}\left(\frac{Vol(S)}{\phi}\right)$
Evolving Set [Andersen-Peres'08]	$\sqrt{\phi \log n}$	$\tilde{O}\left(\frac{Vol(S)}{\sqrt{\phi}}\right)$
Evolving Set [Gharan-Trevisan'12]	$\sqrt{\frac{\phi}{\epsilon}}$	$\tilde{O}\left(\frac{Vol(S)^{1+\epsilon}}{\sqrt{\phi}}\right)$

Cheeger's inequality barrier

Previous results

	Approximation guarantee	Running time
Truncated random walk [Spielman-Teng'04]	$\phi^{\frac{1}{3}} \log^{\frac{2}{3}} n$	$\tilde{O}\left(\frac{Vol(S)}{\phi^{5/3}}\right)$
Truncated random walk [Spielman-Teng'08]	$\sqrt{\phi} \log^{\frac{3}{2}} n$	$\tilde{O}\left(\frac{Vol(S)}{\phi^{2}}\right)$
PageRank random walk [Andersen-Chung-Lang'06]	$\sqrt{\phi \log n}$	$\tilde{O}\left(\frac{Vol(S)}{\phi}\right)$
Evolving Set [Andersen-Peres'08]	$\sqrt{\phi \log n}$	$\tilde{O}\left(\frac{Vol(S)}{\sqrt{\phi}}\right)$
Evolving Set [Gharan-Trevisan'12]	$\sqrt{\frac{\phi}{\epsilon}}$	$\tilde{O}\left(\frac{Vol(S)^{1+\epsilon}}{\sqrt{\phi}}\right)$

Cheeger's inequality barrier

Running time depends only on S and ϕ

Previous results

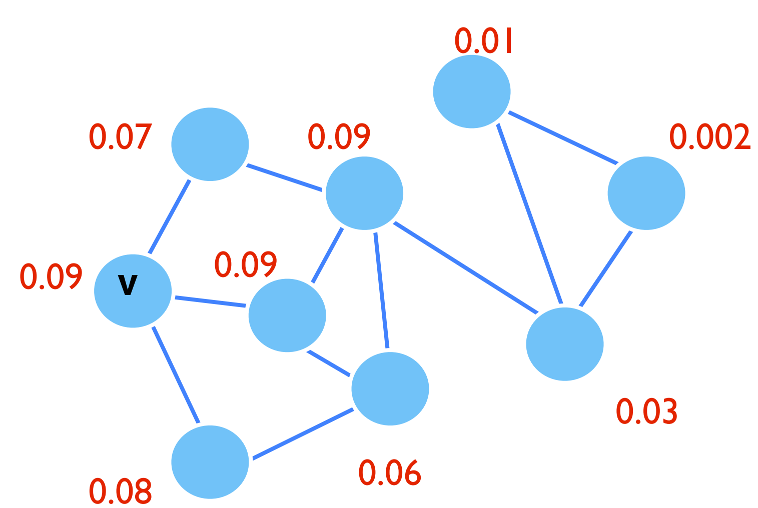
	Approximation guarantee	Running time
Truncated random walk [Spielman-Teng'04]	$\phi^{\frac{1}{3}} \log^{\frac{2}{3}} n$	$\tilde{O}\left(\frac{Vol(S)}{\phi^{5/3}}\right)$
Truncated random walk [Spielman-Teng'08]	$\sqrt{\phi} \log^{\frac{3}{2}} n$	$\tilde{O}\left(\frac{Vol(S)}{\phi^{2}}\right)$
PageRank random walk [Andersen-Chung-Lang'06]	$\sqrt{\phi \log n}$	$\tilde{O}\left(\frac{Vol(S)}{\phi}\right)$
Evolving Set [Andersen-Peres'08]	$\sqrt{\phi \log n}$	$\tilde{O}\left(\frac{Vol(S)}{\sqrt{\phi}}\right)$
Evolving Set [Gharan-Trevisan'12]	$\sqrt{\frac{\phi}{\epsilon}}$	$\tilde{O}\left(\frac{Vol(S)^{1+\epsilon}}{\sqrt{\phi}}\right)$

Cheeger's inequality barrier

Running time depends only on S and ϕ

Clustering using PPR

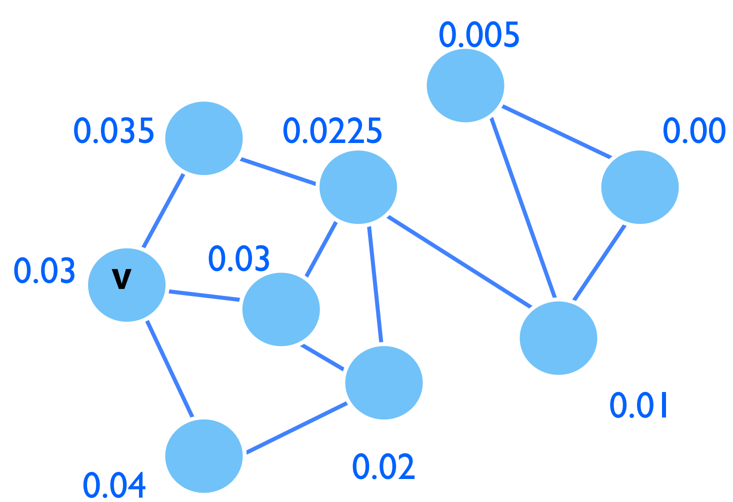
- Approximate Personalized PageRank vector for v



Clustering using PPR

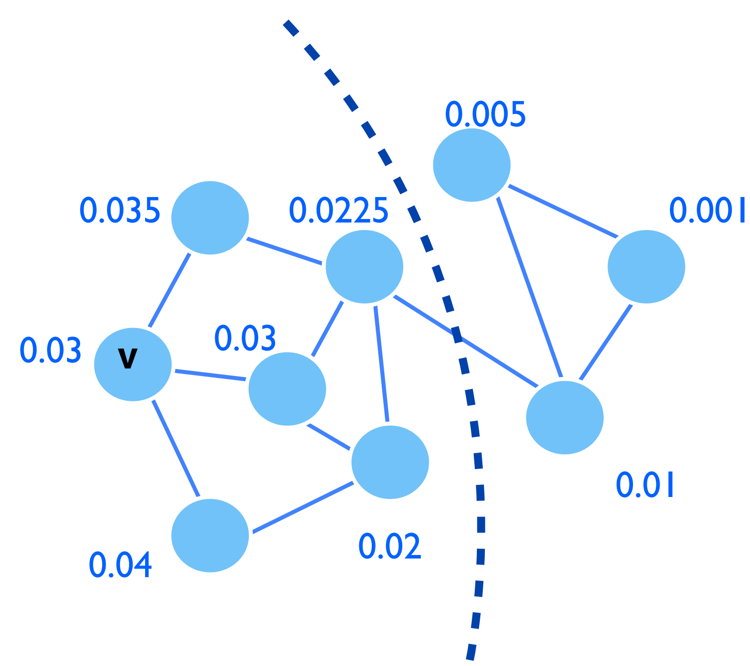
- Approximate Personalized PageRank vector for v
- Sort the nodes according their normalized score

$$\frac{ppr(v,u)}{d(u)}$$



Clustering using PPR

- Approximate Personalized PageRank vector for v
- Sort the nodes according their normalized score
- Select the sweep cut of best conductance



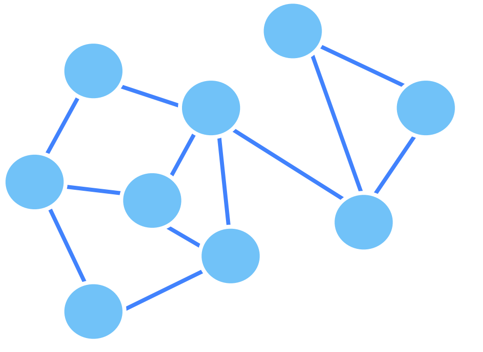
Local clustering beyond Cheeger's barrier

Joint work with:
Vahab Mirrokni (Google Research NY)
Zeyaun Allen Zhu (MIT)
ICML 2013

How should we define a cluster?

Good clusters have cut conductance ϕ

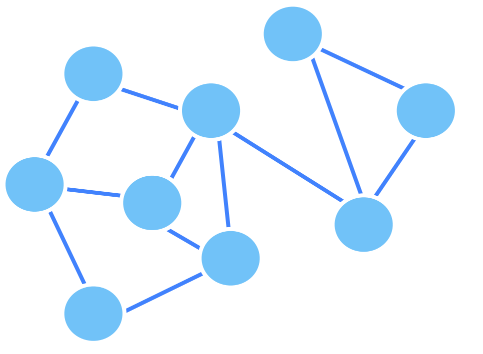
$$\phi = \frac{|cut(C, V - C)|}{min(Vol(C), Vol(V - C))}$$



How should we define a cluster?

Good clusters have cut conductance ϕ

$$\phi = \frac{|cut(C, V-C)|}{min(Vol(C), Vol(V-C))}$$

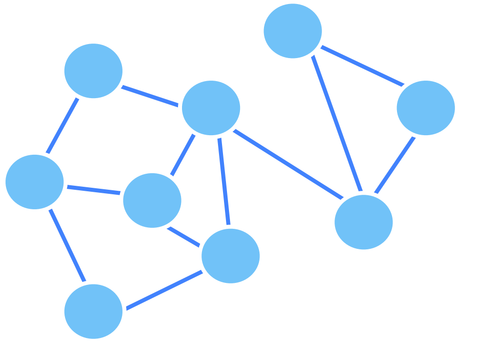


Is it enough to define a good cluster?

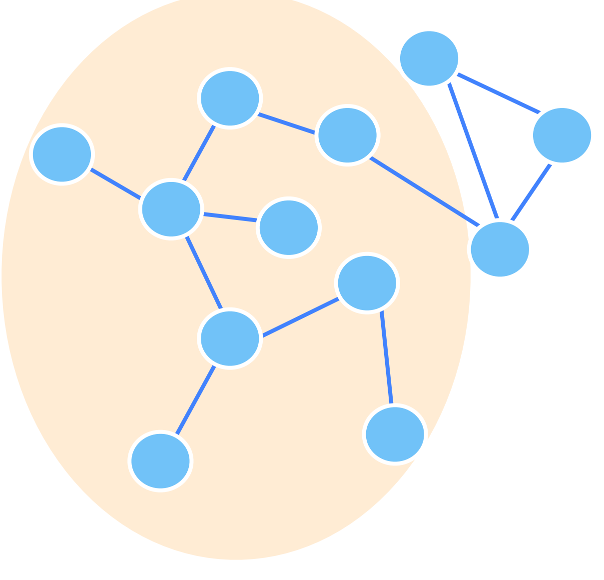
How should we define a cluster?

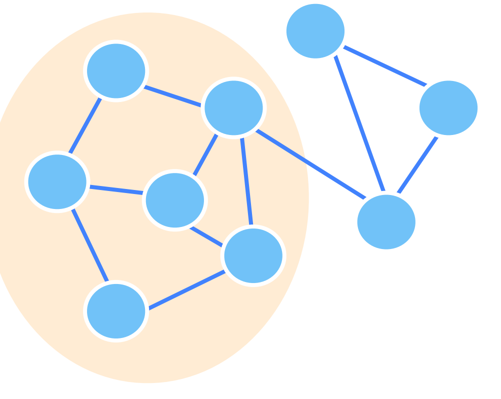
Good clusters have cut conductance ϕ

$$\phi = \frac{|cut(C, V - C)|}{min(Vol(C), Vol(V - C))}$$



Is it enough to define a good cluster?



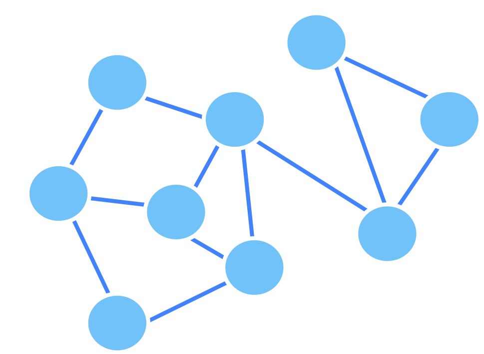


Same cut conductance...

How should we define a cluster?

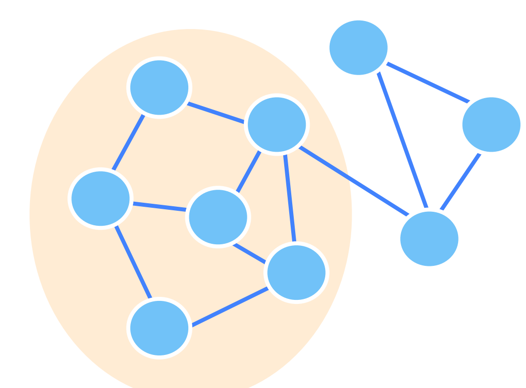
Good clusters have cut conductance ϕ

$$\phi = \frac{|cut(C, V-C)|}{min(Vol(C), Vol(V-C))}$$



Good cluster have good set conductance ψ

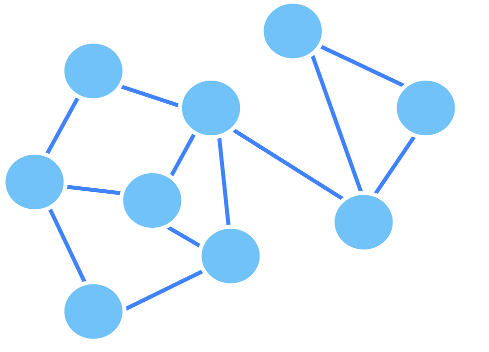
$$\psi = \min_{S \subseteq C} \frac{|cut(S, C-S)|}{min(Vol(S), Vol(C-S))}$$



How should we define a cluster?

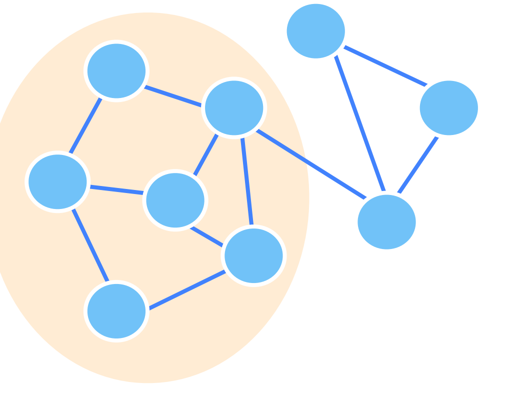
Good clusters have cut conductance ϕ

$$\phi = \frac{|cut(C, V - C)|}{min(Vol(C), Vol(V - C))}$$



Good cluster have good set conductance ψ

$$\psi = \min_{S \subseteq C} \frac{|cut(S, C - S)|}{min(Vol(S), Vol(C - S))}$$



Can we do better when $\psi >> \phi$?

Previous results

	Approximation guarantee	Running time
Truncated random walk [Spielman-Teng'04]	$\phi^{\frac{1}{3}} \log^{\frac{2}{3}} n$	$\tilde{O}\left(\frac{Vol(S)}{\phi^{5/3}}\right)$
Truncated random walk [Spielman-Teng'08]	$\sqrt{\phi} \log^{\frac{3}{2}} n$	$\tilde{O}\left(\frac{Vol(S)}{\phi^{2}}\right)$
PageRank random walk [Andersen-Chung-Lang'06]	$\sqrt{\phi \log n}$	$\tilde{O}\left(\frac{Vol(S)}{\phi}\right)$
Evolving Set [Andersen-Peres'08]	$\sqrt{\phi \log n}$	$\tilde{O}\left(\frac{Vol(S)}{\sqrt{\phi}}\right)$
Evolving Set [Gharan-Trevisan'12]	$\sqrt{\frac{\phi}{\epsilon}}$	$\tilde{O}\left(\frac{Vol(S)^{1+\epsilon}}{\sqrt{\phi}}\right)$

Cheeger's inequality barrier

Running time depends only on S and ϕ

Our hypothesis

We study the problem when $\dfrac{\phi}{\psi^2} < O\left(\dfrac{1}{\log n}\right)$

Our hypothesis

We study the problem when $\frac{\phi}{\psi^2} < O\left(\frac{1}{\log n}\right)$

Similar problem studied Makarychev et al. in STOC12
They assume that

$$\frac{\phi}{\lambda_1} < C$$

give a global SDP that can find communities with cut conductance ϕ

Can we obtain the same results locally?

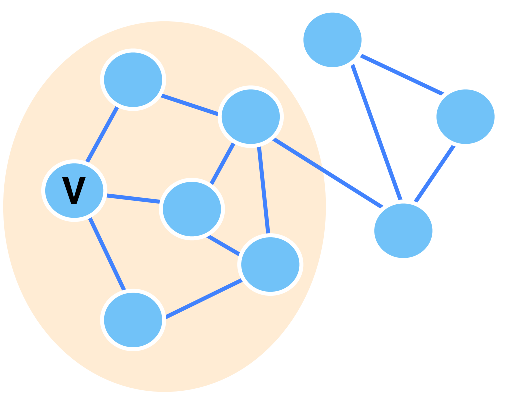
Can we obtain a similar result using the Personalized PageRank?

Theorem

If there is a cluster of cut conductance ϕ and set conductance ψ exists then normalized personalized PageRank find a cluster with conductance $\tilde{o}\left(\frac{\phi}{\psi}\right)$

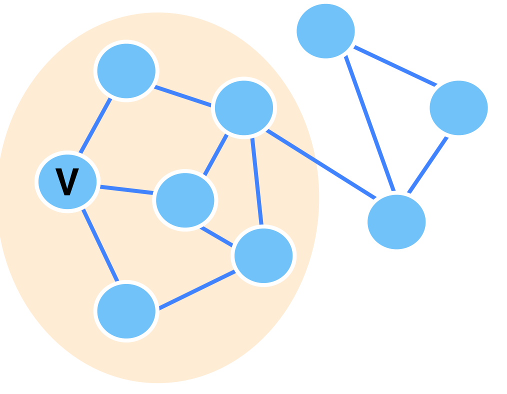
Main proof ideas

Bound the probability of leaving a set in t step knowing that in each step we leave with probability ϕ



Main proof ideas

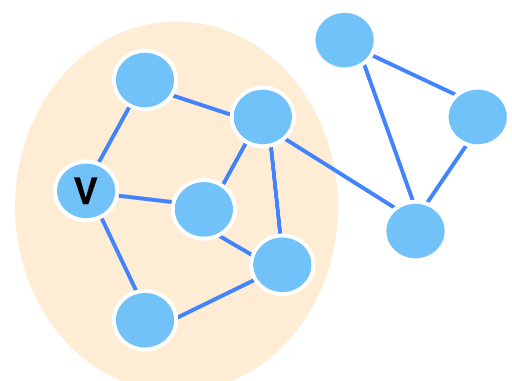
Bound the probability of leaving a set in t step knowing that in each step we leave with probability ϕ



Suppose that we are mixed inside C, then we would leak ϕ probability mass at each step.

Main proof ideas

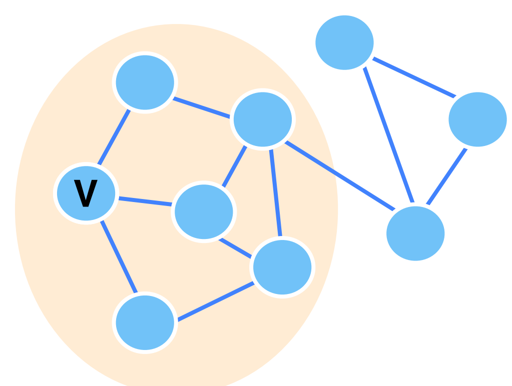
Bound the probability of leaving a set in t step knowing that in each step we leave with probability ϕ



So in $\frac{1}{\alpha}$ steps, we would leak $\frac{\phi}{\alpha}$

Main proof ideas

Bound the probability of leaving a set in t step knowing that in each step we leave with probability ϕ

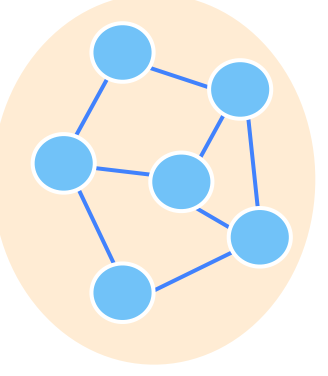


If we start from a good node is:

$$\sum_{u \notin S} pr(u) < \frac{2\phi}{\alpha}$$

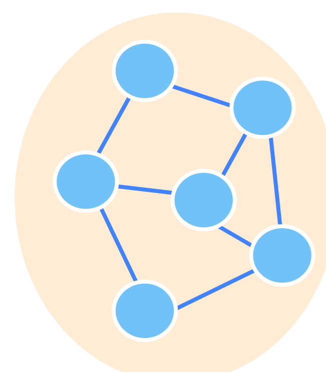
Main proof ideas

Inside S the random walk would be mixed in $\frac{1}{\psi^2}$ steps



Main proof ideas

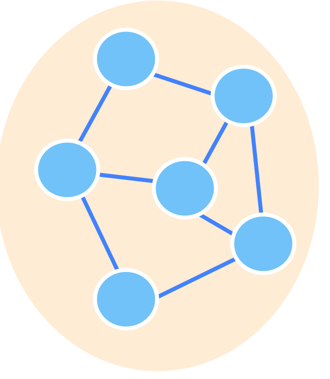
Inside S the random walk would be mixed in $\frac{1}{\psi^2}$ steps



So after $\frac{1}{\psi^2}$ each node would have a score $\frac{d(u)}{Vol(S)}$

Main proof ideas

Inside S the random walk would be mixed in $\frac{1}{\psi^2}$ steps

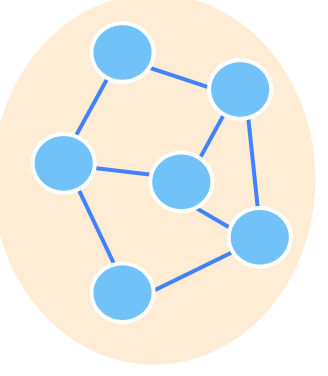


We can express the score of a node inside as:

$$pr(v) \geq \tilde{pr}(v) - pr_l(v)$$

Main proof ideas

Inside S the random walk would be mixed in $\frac{1}{\psi^2}$ steps



We can express the score of a node inside as:

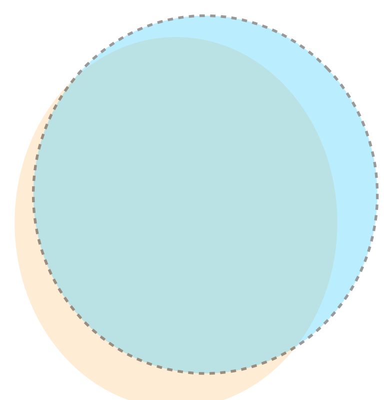
$$pr(v) \geq \tilde{pr}(v) - pr_l(v)$$

But we have a bound:

$$\sum_{v \in S} pr_l(v) = \sum_{z \notin S} ppr(z) \leq 2\frac{\phi}{\psi^2} < O\left(\frac{1}{\log n}\right)$$

Main proof ideas

We can prove that we find a set that partially overlaps with S

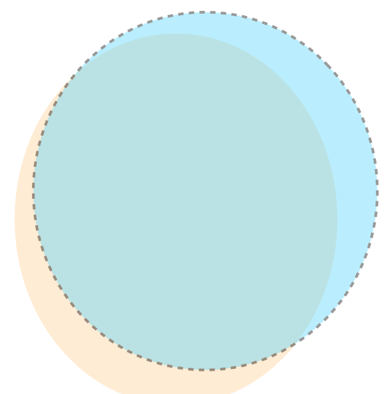


- Most of nodes in the cluster have high score

- Most of nodes outside the cluster have low score

Main proof ideas

We can prove that we find a set that partially overlaps with S



This implies bound on conductance!!

Can we do better?

Theorem 2

If there is a cluster of cut conductance ϕ and set conductance ψ exists then normalized personalized PageRank find a cluster with conductance $\Omega\left(\frac{\phi}{\psi}\right)$

Results

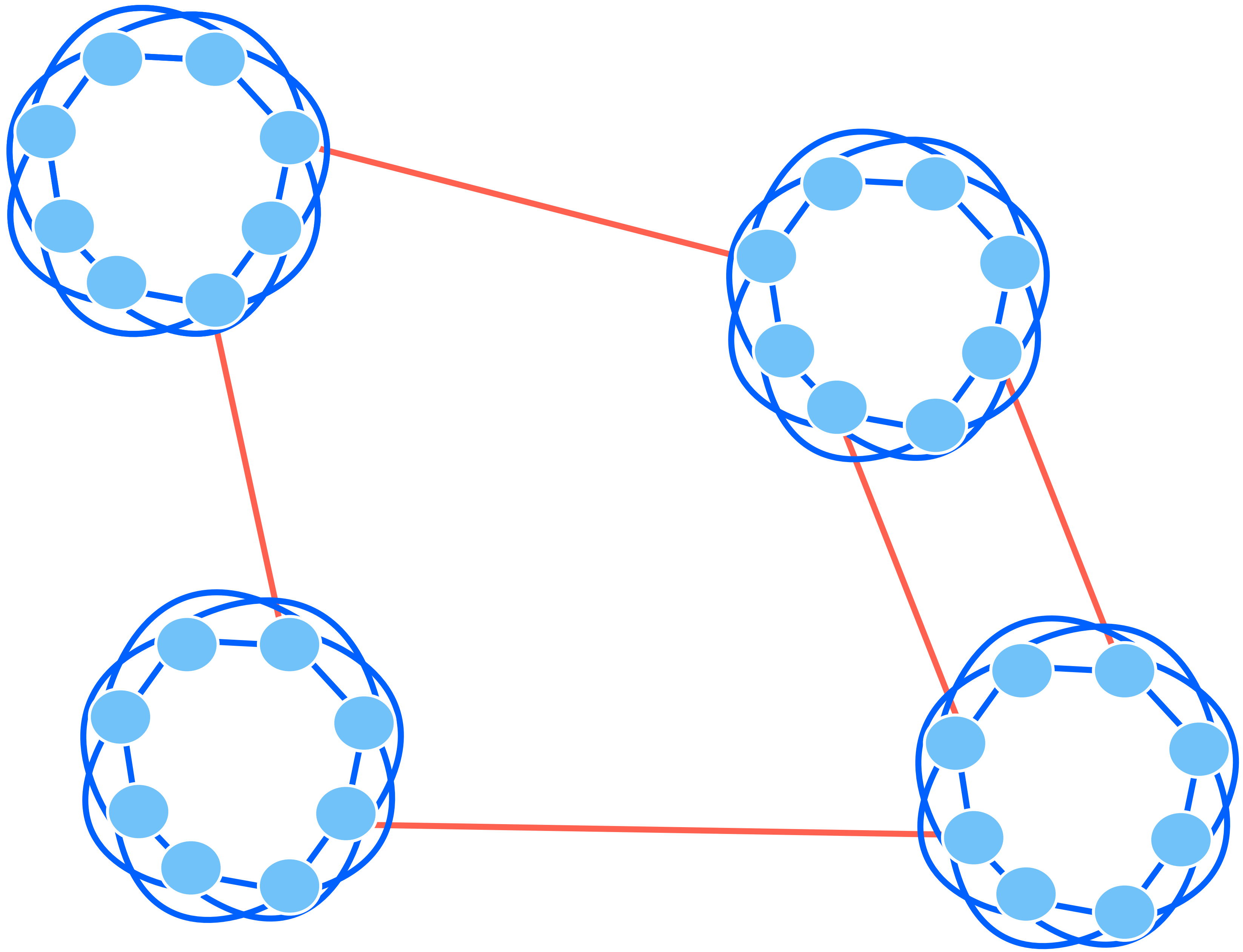
Theorem 1

If there is a cluster of cut conductance ϕ and set conductance ψ exists then normalized personalized PageRank find a cluster with conductance $\tilde{o}\left(\frac{\phi}{\psi}\right)$

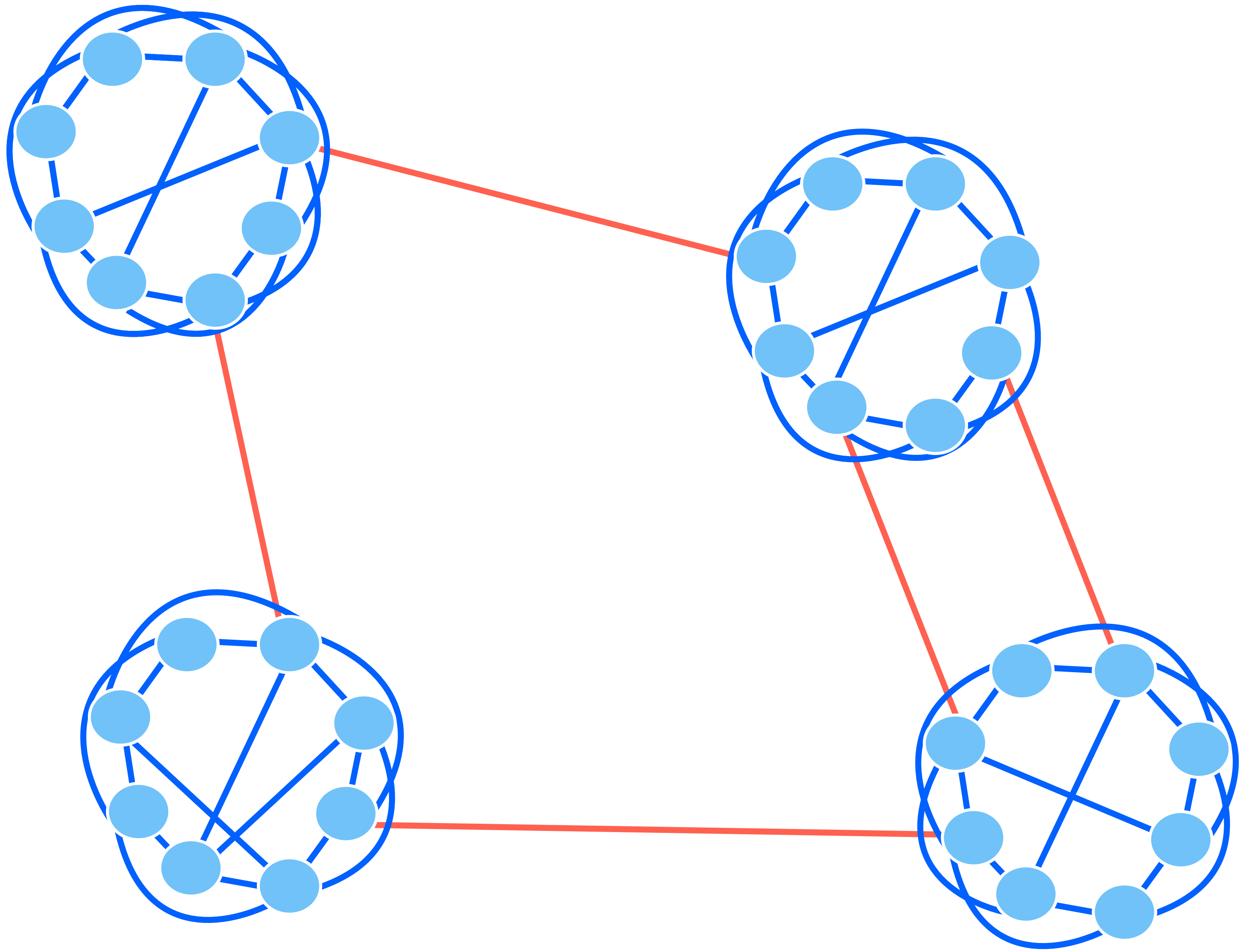
Theorem 2

If there is a cluster of cut conductance ϕ and set conductance ψ exists then normalized personalized PageRank find a cluster with conductance $\Omega\left(\frac{\phi}{\psi}\right)$

Experiments

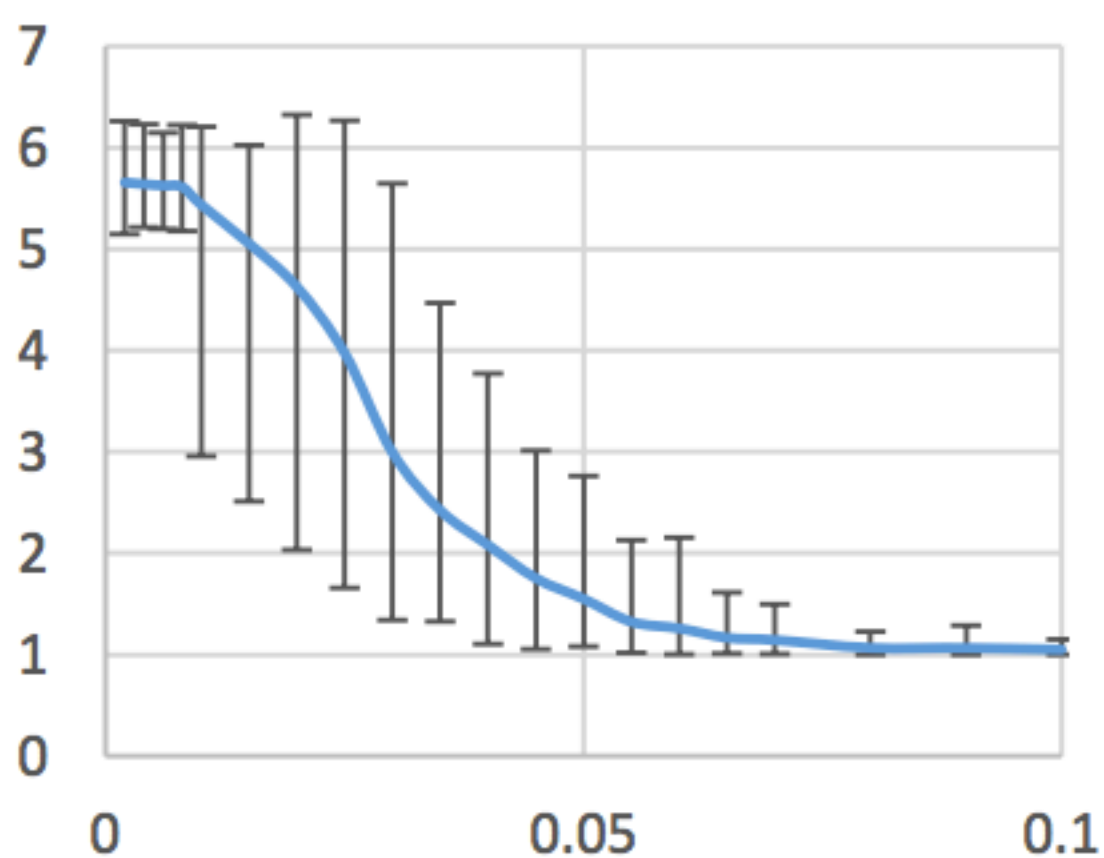


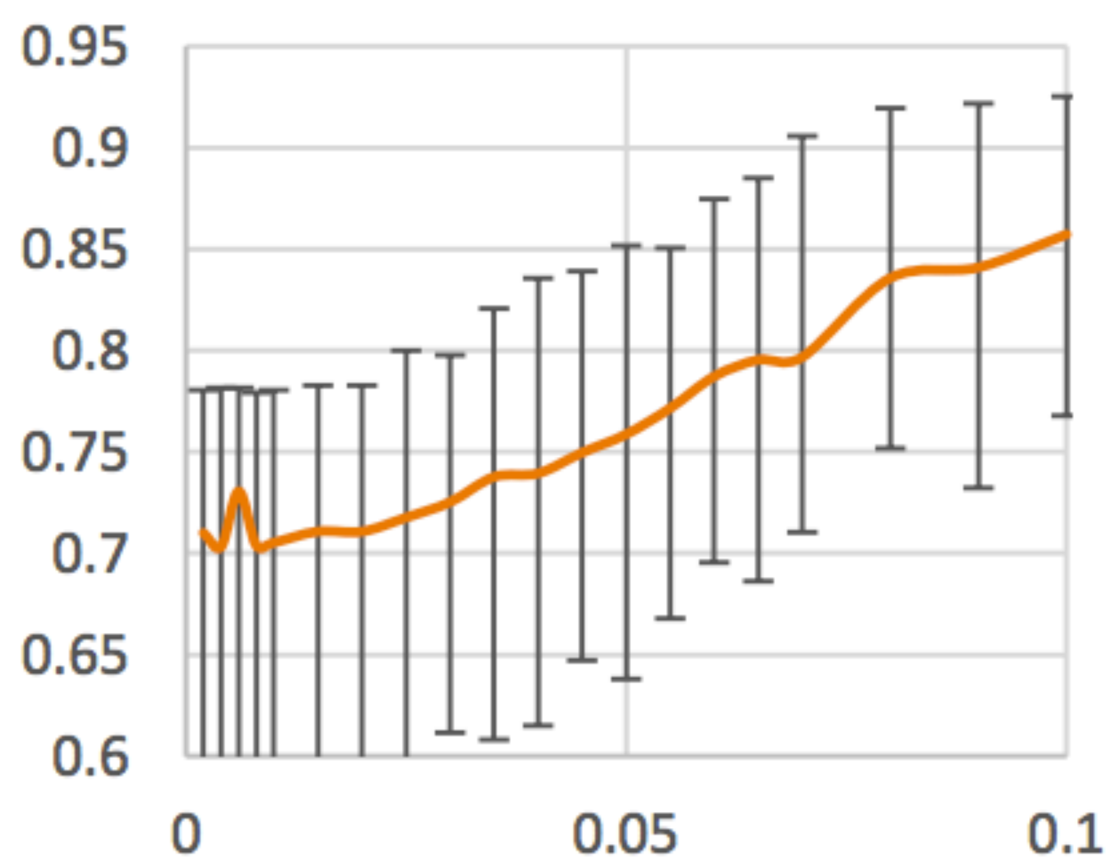
Experiments



Experiments

Experiments using Watts-Strogatz model for the set S





As the gap decreases, precision increases

Conclusion and open problems

Conclusion and open problems

- Random walk based techniques can be used to solve efficiently the similarity and the clustering problem
- Internal connectivity is very important for random walk techniques
- Can we say something when the gap between internal and external connectivity is smaller?

Thanks!