Overlapping Clustering and Distributed Computation

Vahab Mirrokni
Google Research, New York
joint with:
1) R. Khandekar & G. Kortsarz (LATIN),
2) R. Andersen & D. Gleich (WSDM),
3) U. Gargi, W. Lu, & L. Yoon (ICWSM)

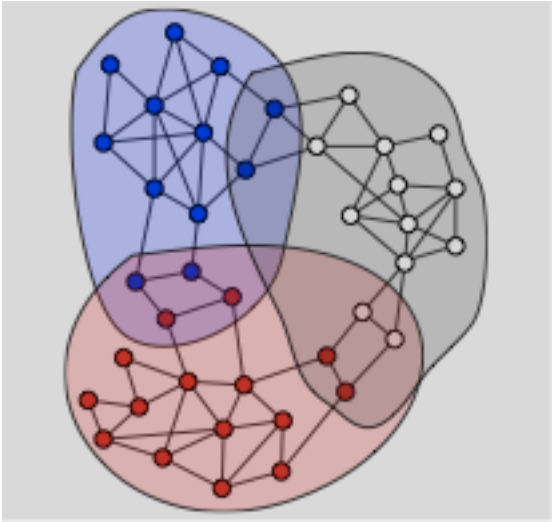
Outline

Overlapping Clustering:

1. Theory: Approximation Algorithms for Minimizing Conductance [with R. Khandekar, G. Kortsarz]
2. Practice: Local Clustering and Large-scale Distributed Clustering
3. Idea: Helping Distributed Computation [with R. Andersen, D. Gleich]

Overlapping Clustering

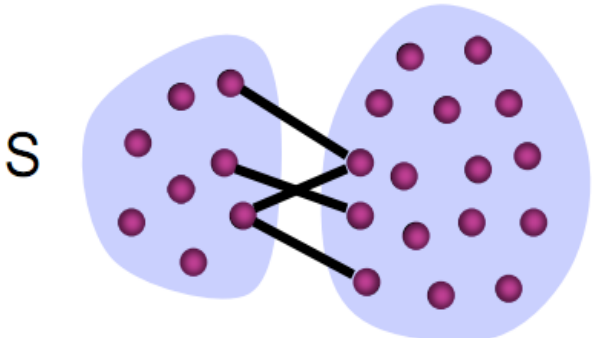
- Motivation:
  1. Natural Social Communities [MSST08,ABL10,…] →
  2. Better clusters (Part 1, KKM)
  3. Easier to compute (distributed) (Part 2, GLMY,AGM)
  4. Useful for Distributed Computation (Part 3, AGM)



- Good Clusters → Low Conductance?
  - Inside: Well-connected,
  - Toward outside: Not so well-connected.

Conductance and Local Clustering

- Conductance of a cluster *S* = $\frac{\#\text{cut edges}}{\min(vol(S), vol(\bar{S}))}$
- Approximation Algorithms
  - *O(log n)*(LR) and $O(\sqrt{\log n})$ (ARV)
- Local Clustering: Given a node *v*, find a min-conductance cluster *S* containing *v*.

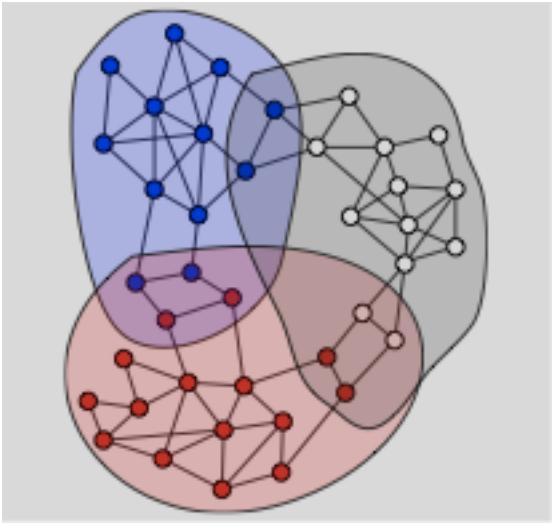


- Local Algorithms based on
  - Truncated Random Walk(ST03), PPR Vectors (ACL07)
  - Empirical study: A cluster with good conductance (LLM10)

Overlapping Clustering

- Find a set of overlapping clusters:

$$\{S_1, \ldots, S_t\}$$



each cluster with volume <= *B*,

covering all nodes,

and minimize:

  - Maximum conductance of clusters (Min-Max)
  - Sum of the conductance of clusters (Min-Sum)

- Overlapping vs. non-overlapping variants?

Overlapping Clustering: Approx. Results

[Khandekar, Kortsarz, M.]

Overlap vs. no-overlap:

- Min-Sum: Within a factor 2 using Uncrossing.
- Min-Max: Might be arbitrarily different.

min-sum	**overlap**	**no-overlap**
bounded-count	Sum.Overlap.Bound $O(\log n)$ (with $O(K)$ clusters)	Sum.Nonoverlap.Bound $O(\log n)$ (with $O(K)$ clusters)
unbounded-count	Sum.Overlap.Unbound $O(\log n)$	Sum.Nonoverlap.Unbound $O(\log n)$

min-max	**overlap**	**no-overlap**
bounded-count	Max.Overlap.Bound $O(\log n)$ (with $O(K \log n)$ clusters)	Max.Nonoverlap.Bound $O(\log^4 n \log\log n)$ (with $O(K)$ clusters)
unbounded-count	Max.Overlap.Unbound $O(\log n)$	Max.Nonoverlap.Unbound $O(\log^4 n \log\log n)$

Outline

Overlapping Clustering:

1. Theory: Approximation Algorithms for Minimizing Conductance
2. Practice: Local Clustering and Large-scale Distributed Overlapping Clustering
3. Idea: Helping Distributed Computation

Local Graph Algorithms

- Local Algorithms: Algorithms based on local message passing among nodes.

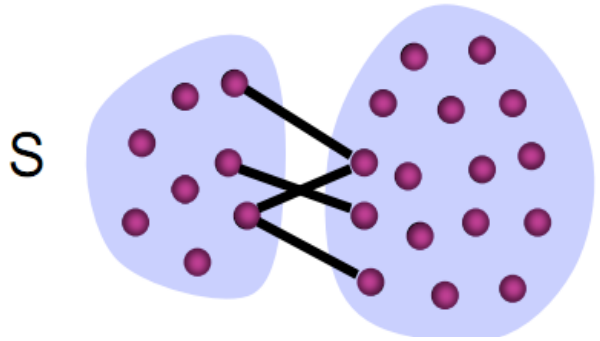
Local Algorithms:

- Applicable in distributed large-scale graphs.
- Faster, Simpler implementation (Mapreduce, Hadoop, Pregel).
- Suitable for incremental computations.

Local Clustering: Recap

- Conductance of a cluster *S* = $\frac{\#\text{cut edges}}{\text{vol}(S)}$

- Goal: Given a node *v*, find a min-conductance cluster *S* containing *v*.



- Local Algorithms based on
  - Truncated Random Walk(ST), PPR Vectors (ACL),

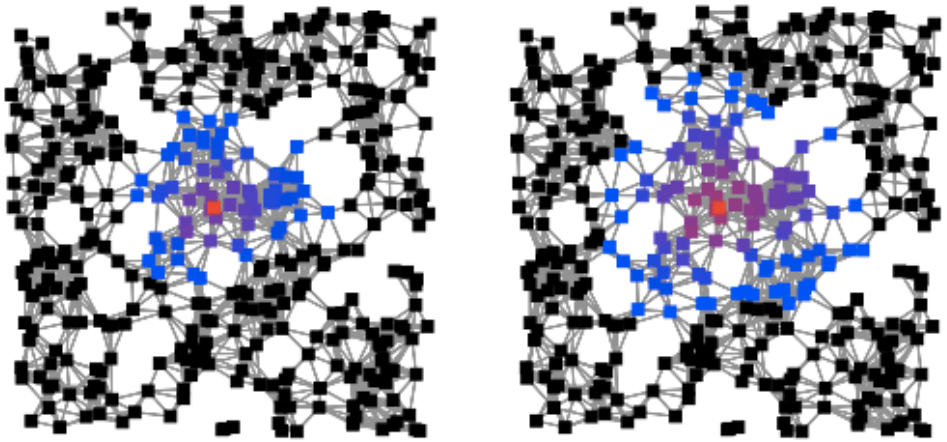
- Outline (Local Distributed Algorithms)
  - Computing Personalized Pagerank Vectors →
  - Local and Overlapping Clustering
  - [Embedding (low-rank matrix approximation)]

Personalized PageRank

- Personalized PageRank (PPR) of $u \rightarrow v$: Probability of visiting v in the following random walk: at each step,
  - With probability α, go back to u.
  - With probability $1-\alpha$, go to a neighbor uniformly at random.
- PPR is a similarity measure: It captures
  - Distance
  - #disjoint paths

Approximate PPR vector

- Personalimized PageRank: Random Walk with Restart.
- PPR Vector for *u*: vector of PPR value from *u*.
- Contribution PR (CPR) vector for *u*: vector of PPR value to *u*.
- Goal: Compute approximate PPR or CPR Vectors with an additive error of ϵ



$\epsilon = .001$ $\epsilon = .0005$

Local PushFlow Algorithms

For each node let $p_v = \vec{0}$ be its initial ppr vector and $r_v = \chi_v$ its residual vector.

- **While** $\max_{u,v} r_v(u) \geq \epsilon$**:**
  - **For each couple of vertices s.t.** $r_v(u) \geq \epsilon$ **:**
    - $p_v(u) = p_v(u) + \alpha r_v(u)$
    - $r_v(u) = r_v(u) + (1-\alpha) r_v(u)/2$
    - **For each** t **such that** $(t,u) \in E$ **:**

      $$r_t(u) = r_t(u) + (1-\alpha) r_v(u)/(2d(u))$$

- **Return the** p_v**vector for each** v**.**

Local Algorithms for PPR

- Local PushFlow Algorithms for approximating both PPR and CPR vectors (ACL07,ABCHMT08)

- Theoretical Guarantees in approximation:
  - Theorem: *O(k)* Push Operations to compute top *k* PPR or CPR values [ACL07,ABCHMT08]

- Simple Pregel or Mapreduce Implementation

PPR-based Local Clustering Algorithm

1. Compute approximate PPR vector for *v*.
2. Sweep(*v*): For each vertex *v*, find the min-conductance set among subsets

$$S_j^{p_v} = \{u_1, \ldots, u_j\}$$

where u_j 's are sorted in the decreasing order of $\frac{p_v(u_j)}{\deg u_j}$.

- Thm[ACL]:If the conductance of the output is ϕ, and the optimum is Φ, then $\phi \leq \sqrt{k\Phi}$ where *k* is the volume of the optimum.

Local Overlapping Clustering

- Modified Algorithm:
  - Find a seed set of nodes that are far from each other.
  - Candidate Clusters: Find a cluster around each node using the local PPR-based algorithms.
  - Solve a covering problem over candidate clusters.
  - Post-process by combining/removing clusters.

- Experiments:
  1. Large-scale Community Detection on Youtube graph (Gargi, Lu, M., Yoon).
  2. On public graphs (Andersen, Gleich, M.)

Large-scale Overlapping Clustering

- Clustering a Youtube video subgraph (Lu, Gargi, M., Yoon, ICWSM 2011)
  - Clustered graphs with 120M nodes and 2B edges in 5 hours.
  - https://sites.google.com/site/ytcommunity
- Overlapping clusters for Distributed Computation (Andersen, Gleich, M.)
  - Ran on graphs with up to 8 million nodes.
  - Compared with Metis and GRACLUS → Better quality (up to 40%) → See next section.

Future Directions

- Design practical algorithms for overlapping clustering with good theoretical guarantees
- Maximize minimum Density?
- Local algorithm for low-rank embedding of large graphs → [Useful for online clustering]
  - Message-passing-based low-rank matrix approximation
  - Ran on a graph with 50M nodes and in 3 hours (using 1000 machines)
  - With Keshavan, Thakur.

Outline

Overlapping Clustering:

1. Theory: Approximation Algorithms for Minimizing Conductance
2. Practice: Local Clustering and Large-scale Distributed Overlapping Clustering
3. Idea: Helping Distributed Computation

Clustering for Distributed Computation

- Implement scalable distributed algorithms
  - Partition the graph → assign clusters to machines
  - must address communication among machines
  - close nodes should go to the same machine
- Idea: Overlapping clusters [Andersen, Gleich, M.]
- Given a graph ***G***, overlapping clustering ***(C, y)*** is
  - a set of clusters ***C*** each with volume < ***B*** and
  - a mapping from each node ***v*** to a home cluster ***y(v)***.
- Message to an outside cluster for ***v*** goes to ***y(v)***.
  - Communication: e.g PushFlow to outside clusters

Formal Metric: Swapping Probability

- In a random walk on an overlapping clustering, the walk moves from cluster to cluster.
- On leaving a cluster, it goes to the home cluster of the new node.
- Swap: A transition between clusters
  - requires a communication if the underlying graph is distributed.
- Swapping Probability := probability of swap in a long random walk.

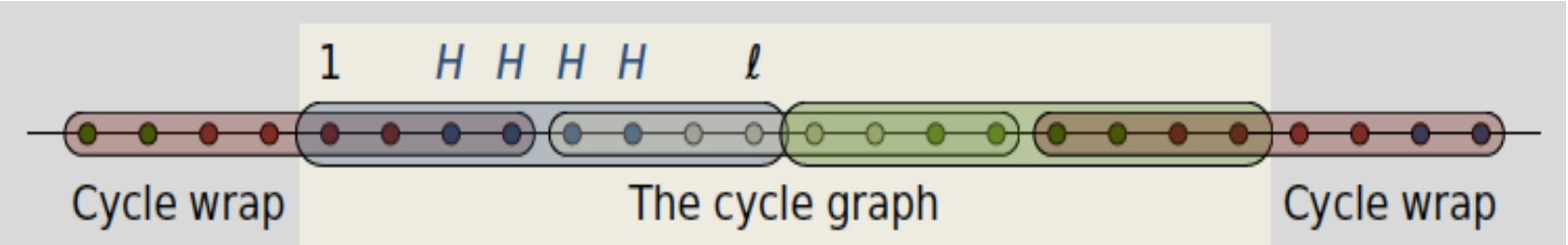
Swapping Probability: Lemmas

- Lemma 1: Swapping Probability for Partitioning $\mathcal{P}$: $\frac{1}{\mathrm{Vol}(G)} \sum_{C \in \mathcal{P}} |\delta(C)|$

- Lemma 2: Optimal swapping probability for overlapping clustering might be arbitrarily better than swapping partitioning.
  - Cycles, Paths, Trees, etc

Lemma 2: Example

- Consider cycle C_n with $n = MB$ nodes.
- Partitioning: *2/B* (*M* paths of volume *B*←Lemma 1)
- Overlapping Clustering: Total volume:*4n=4MB*



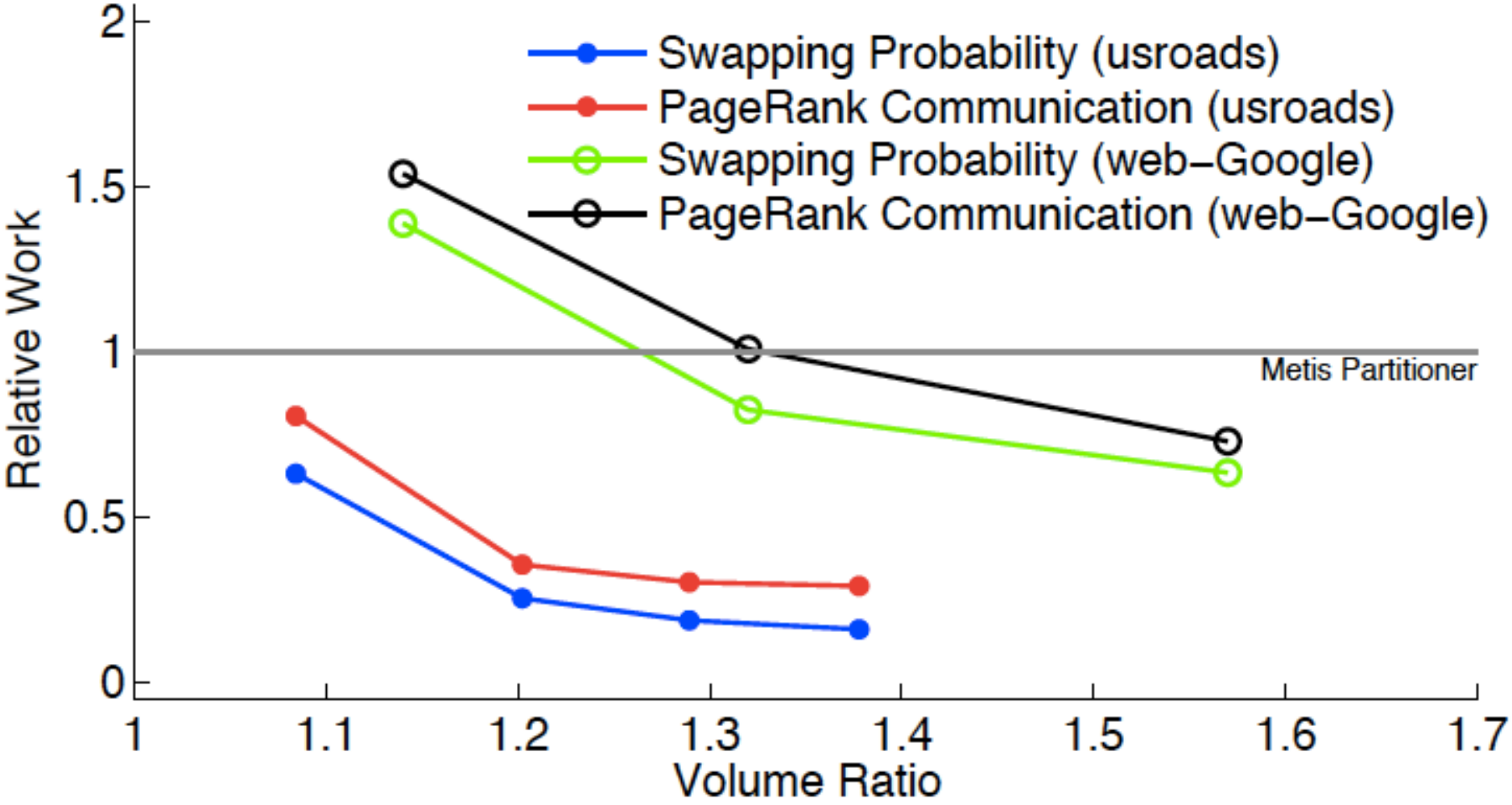
  - When the walk leaves a cluster, it goes to the center of another cluster.
  - A random walk travels $O(\sqrt{t})$ in *t* steps → it takes *B^2/2* to leave a cluster after a swap.
  - → Swapping Probability = *4/B^2*.

Experiments: Setup

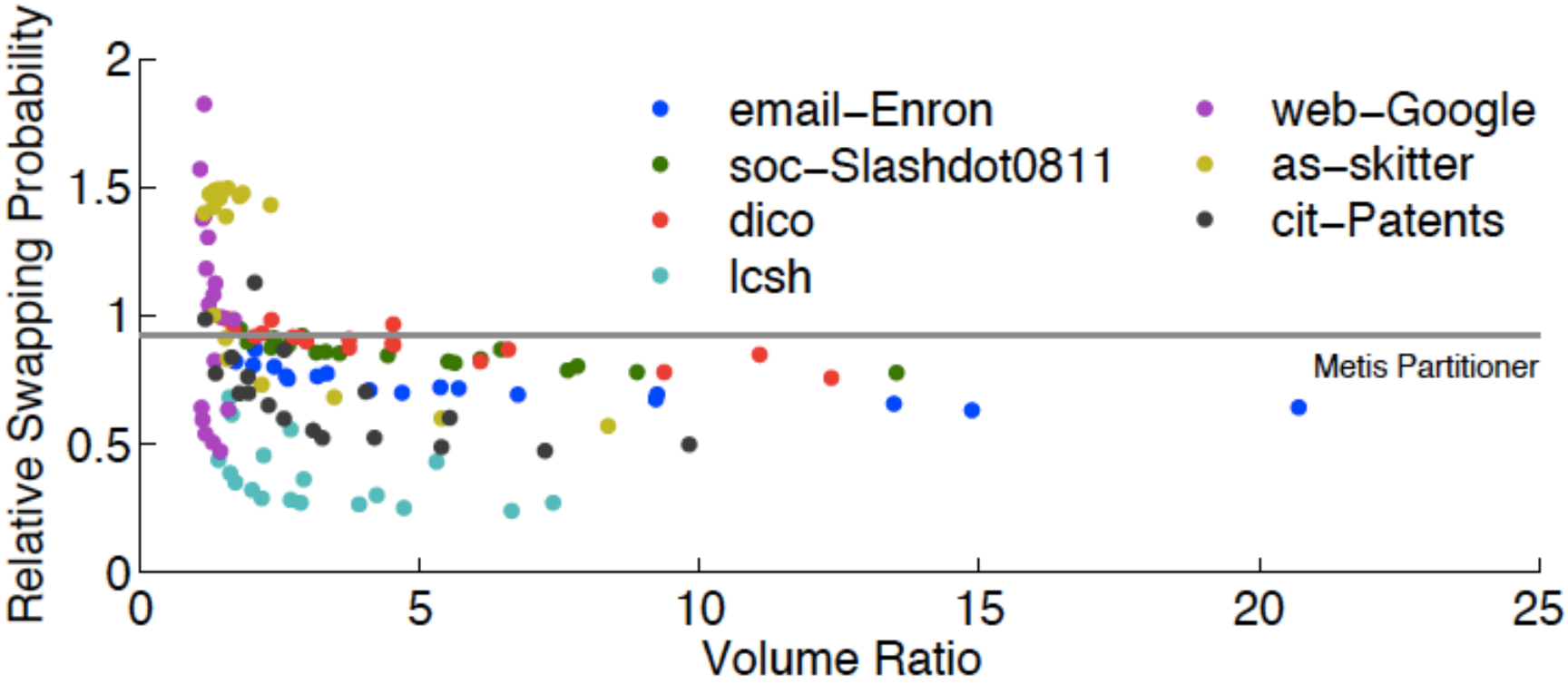
- We empirically study this idea.
- Used overlapping local clustering…
- Compared with Metis and GRACLUS.

Graph	$\|V\|$	$\|E\|$	max deg	$\|E\|/\|V\|$
onera	85567	419201	5	4.9
usroads	126146	323900	7	2.6
annulus	500000	2999258	19	6.0
email-Enron	33696	361622	1383	10.7
soc-Slashdot	77360	1015667	2540	13.1
dico	111982	2750576	68191	24.6
lcsh	144791	394186	1025	2.7
web-Google	855802	8582704	6332	10.0
as-skitter	1694616	22188418	35455	13.1
cit-Patents	3764117	33023481	793	8.8

Swapping Probability and Communication



Swapping Probability



Swapping Probability, Conductance and Communication

Swapping Probability

Graph	Swap. Prob. of Partition	Avg. Cond.	Swap. Prob. of Overlap	Perf. Ratio	Vol. Ratio	Avg. Cond.	Method
onera	7.6×10^{-4}	0.02	1×10^{-4}	0.129	2.82	0.03	Med.30
usroads	1.3×10^{-4}*	<0.01	1×10^{-6}	0.008	1.49	0.01	Med.30
annulus	1×10^{-4}	<0.01	5×10^{-6}	0.049	1.17	<0.01	Med.10
email-Enron	0.02	0.39	0.013	0.650	14.86	0.47	Big.30
soc-Slashdot	0.03	0.66	0.026	0.867	13.52	0.65	Med.30
dico	0.04	0.82	0.03	0.750	12.35	0.82	Big.30
lcsh	0.003*	0.06	0.0007	0.233	6.63	0.12	Med.30
web-Google	7.8×10^{-4}*	0.02	4.6×10^{-4}	0.592	1.43	0.02	Big.30
as-skitter	0.005	0.1	0.004	0.549	8.36	0.2	Big.30
cit-Patents	0.0064	0.13	0.0034	0.524	3.25	0.42	Small.10

Communication

Graph	Comm. of Partition	Avg. Cond.	Comm. of Overlap	Perf. Ratio	Vol. Ratio	Avg. Cond.	Method
onera	18654	0.02	48	0.003	2.82	0.03	Med.30
usroads	3256*	<0.01	0	0.000	1.49	0.01	Med.30
annulus	12074	<0.01	2	0.000	0.01	<0.01	Med.10
email-Enron	194536*	0.4	235316	1.210	1.7	0.46	Metis.2
soc-Slashdot	875435*	0.68	1.3×10^6	1.480	1.78	0.74	Metis.2
dico	1.5×10^6*	0.79	2.0×10^6	1.320	1.53	0.84	Metis.2
lcsh	73000*	0.06	48777	0.668	2.17	0.08	Small.5
web-Google	201159*	0.02	167609	0.833	1.57	0.04	Metis.10
as-skitter	2.4×10^6	0.1	3.9×10^6	1.645	1.93	0.24	Metis.10
cit-Patents	8.7×10^6	0.13	7.3×10^6	0.845	1.34	0.16	Metis.4

A challenge and an idea

- Challenge: To accelerate the distributed implementation of local algorithms, close nodes (clusters) should go to the same machine ← Chicken or Egg Problem.
- Idea: Use Overlapping clusters:
  - Simpler for preprocessing.
  - Improve communication cost (Andersen, Gleich, M.)
- Apply the idea iteratively?

Open Problems

- Practical algorithms with good theoretical guarantees
- Maximize minimum Density?
- Design approximation algorithms for swapping probability metric?
- Classify graphs in which overlapping clustering helps in getting a much better swapping probability.
- How do we solve the chicken or egg problem?

Thank You!

Thanks

Message-Passing-based Embedding

- Let $N \in \mathbb{R}^{m \times n}$ be 'approximately' low rank

$$N = UV^T + W$$

- A small subset E of entries revealed
- U and V are typically low rank
- Compute $\widehat{UV^T}$ from the subset of entries revealed

- Pregel Implementation of Message-passing-based low-rank matrix approximation.
- Ran on G+ graph with 40 million nodes and used for friend suggestion: Better link prediction than PPR.

Overlap vs. no-overlap

- Min-Sum: Overlap is within a factor 2 of no-overlap. This is done through uncrossing:
  - $(X, Y) \rightarrow$ either $(X, Y \backslash X)$ or $(Y, X \backslash Y)$
- Min-Max: For a family of graphs, min-max solution is very different for overlap vs. no-overlap:
  - For Overlap, it is $O(|V|^{-2/3})$.
  - For no-overlap is $\Omega(1)$.

Overlap vs. no-overlap: Min-Max

- Min-Max: For some graphs, min-max conductance from overlap << no-overlap.
  - For an integer *k*, let $G = K_k \cup H$, where *H* is a 3-regular expander on k^3 nodes, and $B = k(k-1)+3$.
  - Overlap: for each $v \in H$, $C_v = K_k \cup \{v\}$, thus min-max conductance $O(|V|^{-2/3})$
  - Non-overlap: Conductance of at least one cluster is at least $\Omega(1)$, since *H* is an expander.

Overlapping Clustering: Basic Idea

- Basic Framework:
  1. Find a candidate set of clusters around nodes.
  2. Run a greedy set covering algorithm and choose a subset of candidate clusters covering all nodes.
- Challenge in applying set-cover algorithm: Find a good candidate set of clusters and find the cluster with the maximum size/cost ratio?
- Racke: Embed the graph into a family of trees while preserving the cut value.

Tree Embedding and Dynamic Program

- Racke: For any graph *G(V,E)*, there exists an embedding of *G* to a convex combination of trees T_i such that the value of each cut is preserved within a $\log n$ factor in expectation.

→ Implement set-cover algorithm over trees.

- In order to find the most cost-effective cluster, run a dynamic program over the tree.

Overlapping Clustering: Approx. Results

Overlap vs. no-overlap:

- Min-Sum: Within a factor 2 using Uncrossing.
- Min-Max: Might be arbitrarily different.

min-sum	**overlap**	**no-overlap**
bounded-count	Sum.Overlap.Bound $O(\log n)$ (with $O(K)$ clusters)	Sum.Nonoverlap.Bound $O(\log n)$ (with $O(K)$ clusters)
unbounded-count	Sum.Overlap.Unbound $O(\log n)$	Sum.Nonoverlap.Unbound $O(\log n)$

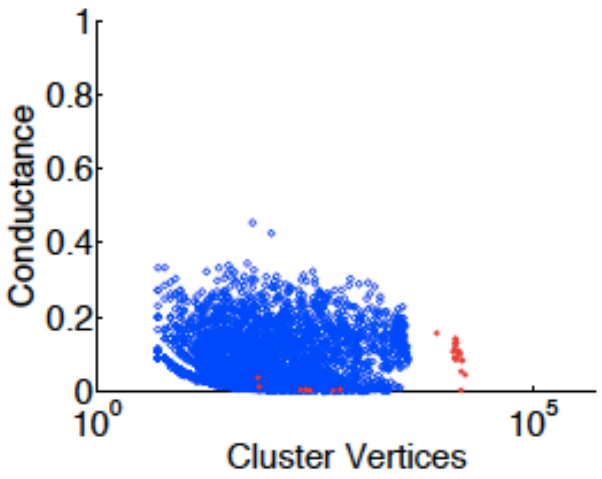
min-max	**overlap**	**no-overlap**
bounded-count	Max.Overlap.Bound $O(\log n)$ (with $O(K \log n)$ clusters)	Max.Nonoverlap.Bound $O(\log^4 n \log\log n)$ (with $O(K)$ clusters)
unbounded-count	Max.Overlap.Unbound $O(\log n)$	Max.Nonoverlap.Unbound $O(\log^4 n \log\log n)$

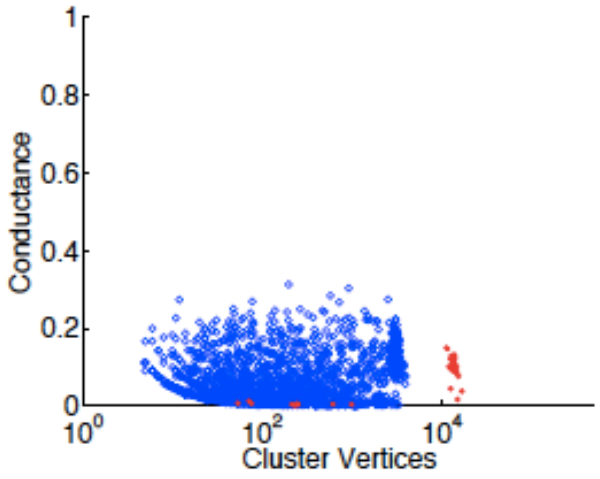
Experiments: Public Data

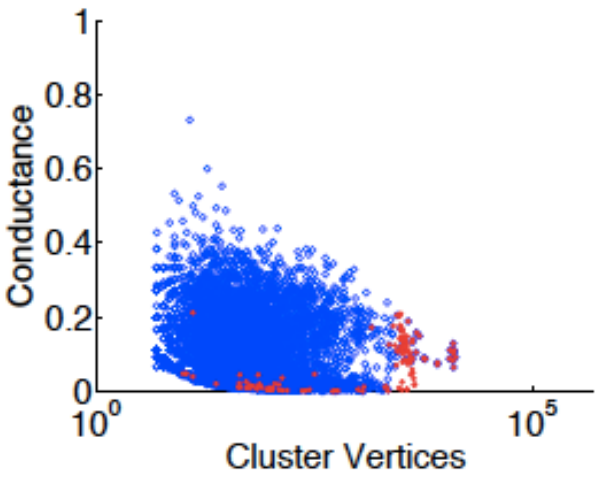
Graph	$\|V\|$	$\|E\|$	max deg	$\|E\|/\|V\|$
onera	85567	419201	5	4.9
usroads	126146	323900	7	2.6
annulus	500000	2999258	19	6.0
email-Enron	33696	361622	1383	10.7
soc-Slashdot	77360	1015667	2540	13.1
dico	111982	2750576	68191	24.6
lcsh	144791	394186	1025	2.7
web-Google	855802	8582704	6332	10.0
as-skitter	1694616	22188418	35455	13.1
cit-Patents	3764117	33023481	793	8.8

Average Conductance

- Goal: get clusters with low conductance and volume up to 10% of total volume
- Start from various sizes and combine.
  - Small clusters: up to volume 1000
  - Medium clusters: up to volume 10000
  - Large Clusters: up to 10% of total volume.







Impact of Heuristic: Combining Clusters