Overlapping Community Detection Using Seed Set Expansion

Joyce Jiyoung Whang[1] David F. Gleich[2] Inderjit S. Dhillon[1]

[1]The University of Texas at Austin

[2]Purdue University

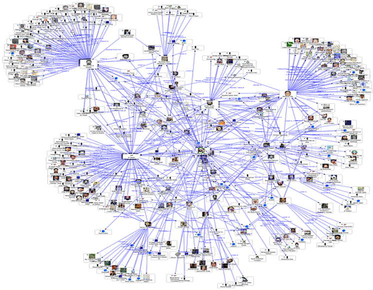
International Conference on Information and Knowledge Management

Oct. 27th - Nov. 1st, 2013.

Contents

Overlapping Communities

- Community (cluster) in a graph $G = (\mathcal{V}, \mathcal{E})$
  - Set of cohesive vertices
  - Communities naturally overlap (e.g. social circles)
- Graph Clustering (Partitioning)
  - k disjoint clusters $\mathcal{C}_1, \cdots, \mathcal{C}_k$ such that $\mathcal{V} = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_k$
- Overlapping Community Detection
  - k overlapping clusters such that $\mathcal{C}_1 \cup \cdots \cup \mathcal{C}_k \subseteq \mathcal{V}$



Real-world Networks

- Collaboration networks: co-authorship
- Social networks: friendship
- Product network: co-purchasing information

Graph	No. of vertices	No. of edges
Collaboration networks		
HepPh	11,204	117,619
AstroPh	17,903	196,972
CondMat	21,363	91,286
DBLP	317,080	1,049,866
Social networks		
Flickr	1,994,422	21,445,057
Myspace	2,086,141	45,459,079
LiveJournal	1,757,326	42,183,338
Product network		
Amazon	334,863	925,872

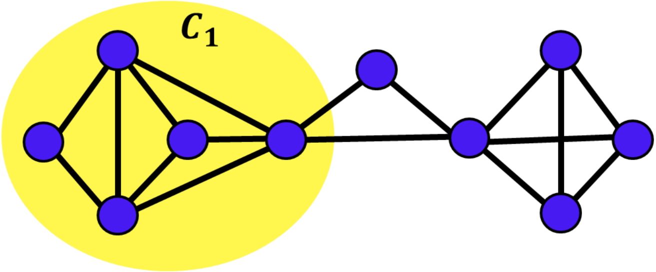
Measures of cluster quality

- Normalized Cut of a cluster

$$\texttt{ncut}(\mathcal{C}_i) = \frac{\texttt{links}(\mathcal{C}_i, \mathcal{V} \backslash \mathcal{C}_i)}{\texttt{links}(\mathcal{C}_i, \mathcal{V})}.$$

- Conductance

$$\texttt{conductance}(\mathcal{C}_i) = \frac{\texttt{links}(\mathcal{C}_i, \mathcal{V} \backslash \mathcal{C}_i)}{\min\left(\texttt{links}(\mathcal{C}_i, \mathcal{V}), \texttt{links}(\mathcal{V} \backslash \mathcal{C}_i, \mathcal{V})\right)}.$$



Graph Clustering and Weighted Kernel k-Means

- **A general weighted kernel k-means objective** is equivalent to **a weighted graph clustering objective** (Dhillon et al. 2007).
- Weighted kernel k-means
  - Objective

$$J = \sum_{c=1}^{k} \sum_{\mathbf{x}_i \in \pi_c} w_i ||\varphi(\mathbf{x}_i) - \mathbf{m}_c||^2, \text{ where } \mathbf{m}_c = \frac{\sum_{\mathbf{x}_i \in \pi_c} w_i \varphi(\mathbf{x}_i)}{\sum_{\mathbf{x}_i \in \pi_c} w_i}.$$

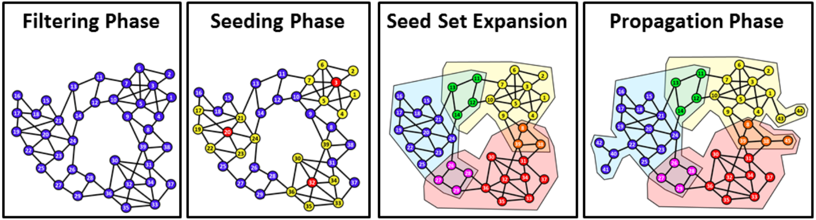
- Distance between a vertex $v \in \mathcal{C}_i$ and cluster $\mathcal{C}_i$

$$\mathtt{dist}(v, \mathcal{C}_i) = -\frac{2\,\mathtt{links}(v, \mathcal{C}_i)}{\mathtt{deg}(v)\,\mathtt{deg}(\mathcal{C}_i)} + \frac{\mathtt{links}(\mathcal{C}_i, \mathcal{C}_i)}{\mathtt{deg}(\mathcal{C}_i)^2} + \frac{\sigma}{\mathtt{deg}(v)} - \frac{\sigma}{\mathtt{deg}(\mathcal{C}_i)}$$

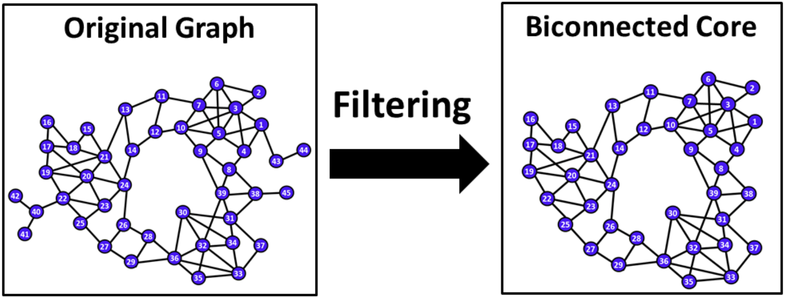
The Proposed Algorithm

Proposed Algorithm

- Seed Set Expansion
  - Carefully select seeds
  - Greedily expand communities around the seed sets
- The algorithm
  - Filtering Phase
  - Seeding Phase
  - Seed Set Expansion Phase
  - Propagation Phase

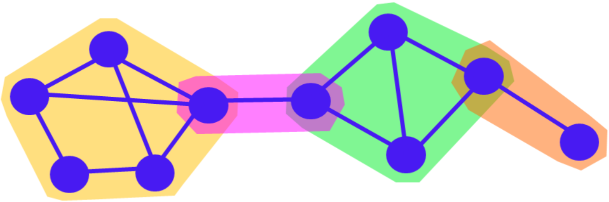


Filtering Phase

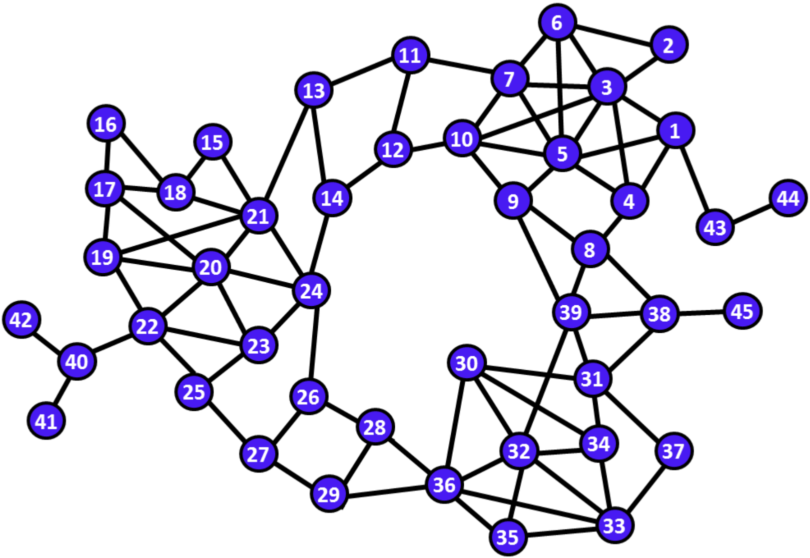


Filtering Phase

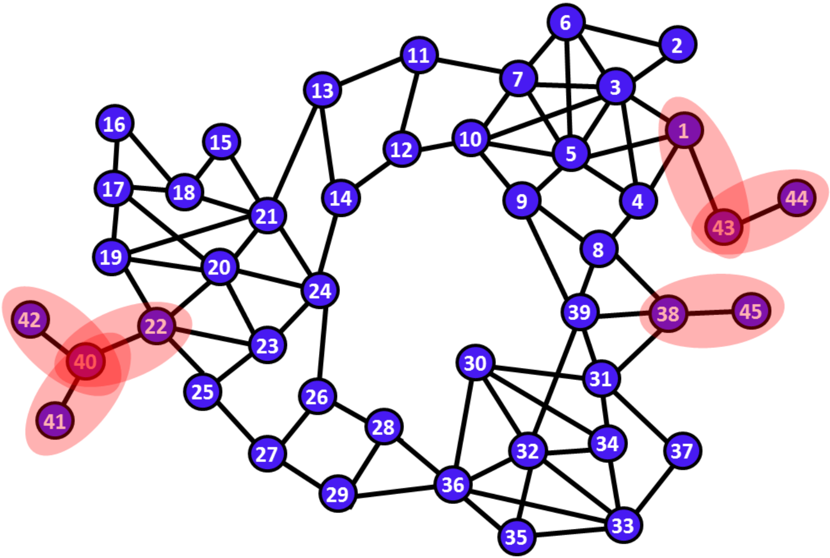
- Remove unimportant regions of the graph
  - Trivially separable from the rest of the graph
  - Do not participate in overlapping clustering
- Our filtering procedure
  - Remove all single-edge biconnected components (remain connected after removing any vertex and its adjacent edges)
  - Compute the largest connected component (LCC)



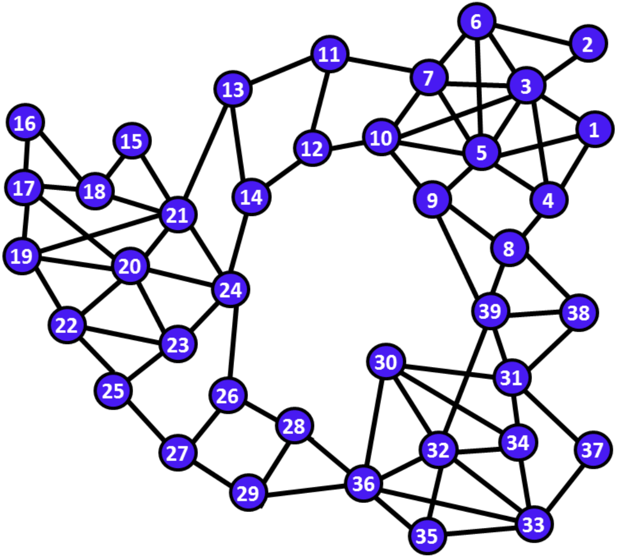
Filtering Phase



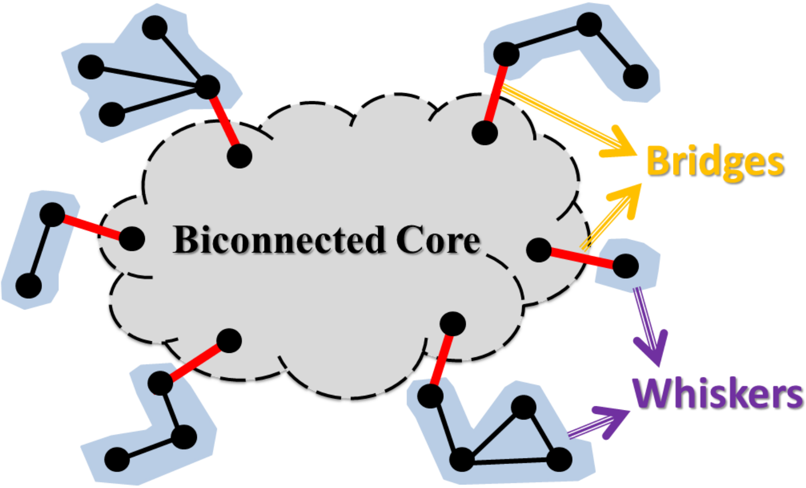
Filtering Phase



Filtering Phase



Filtering Phase

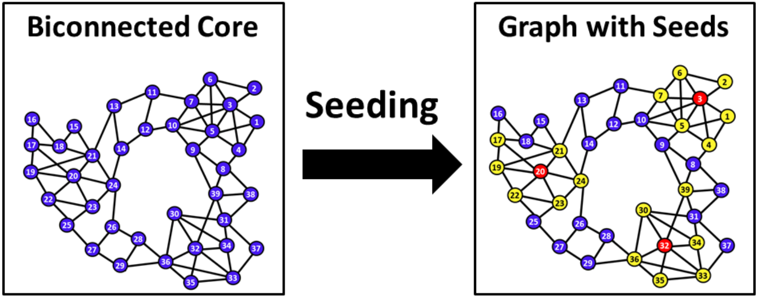


Filtering Phase

	Biconnected core		Detached graph	
	No. of vertices (%)	No. of edges (%)	No. of components	Size of LCC (%)
HepPh	9,945 (88.8%)	116,099 (98.7%)	1,123	21 (0.0019%)
AstroPh	16,829 (94.0%)	195,835 (99.4%)	957	23 (0.0013%)
CondMat	19,378 (90.7%)	89,128 (97.6%)	1,669	12 (0.00056%)
DBLP	264,341 (83.4%)	991,125 (94.4%)	43,093	32 (0.00010%)
Flickr	954,672 (47.9%)	20,390,649 (95.1%)	864,628	107 (0.000054%)
Myspace	1,724,184 (82.7%)	45,096,696 (99.2%)	332,596	32 (0.000015%)
LiveJournal	1,650,851 (93.9%)	42,071,541 (99.7%)	101,038	105 (0.000060%)
Amazon	291,449 (87.0%)	862,836 (93.2%)	25,835	250 (0.00075%)

- The biconnected core – substantial portion of the edges
- Detached graph – likely to be disconnected
- Whiskers – separable from each other, no significant size

Seeding Phase



Seeding Phase

- Graclus centers
  - Graclus: a high quality and efficient graph partitioning scheme

---

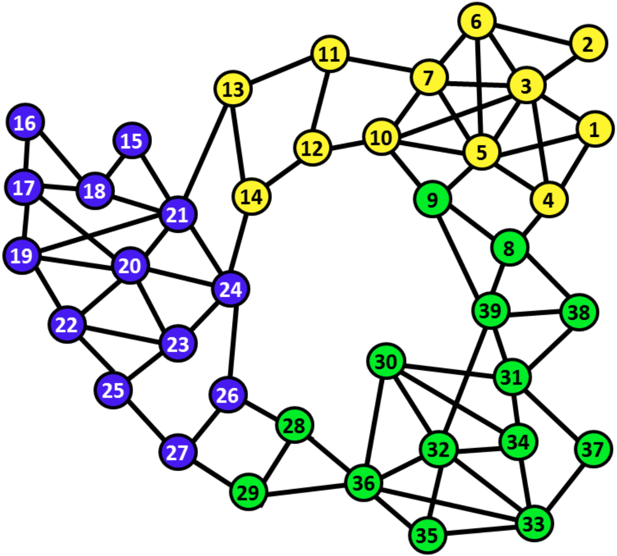
Algorithm 1 Seeding by Graclus Centers

Input: graph G, the number of seeds k.
Output: the seed set $\mathcal{S}$.

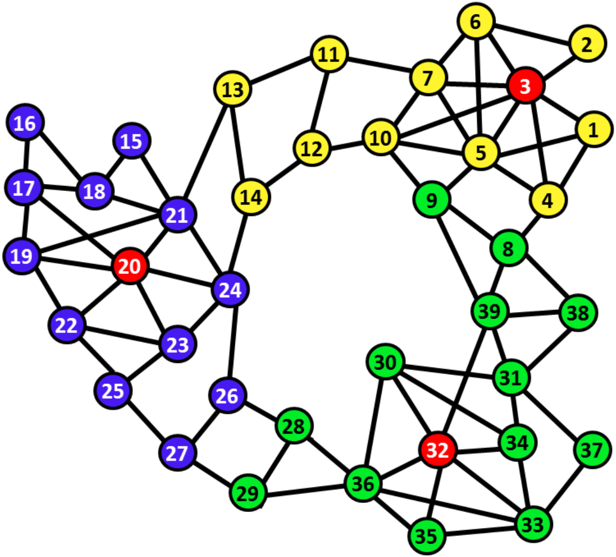
```
Compute exhaustive and non-overlapping clusters C_i (i=1, ..., k) on G.
Initialize S = ∅.
for each cluster C_i do
    for each vertex v ∈ C_i do
        Compute dist(v, C_i).
    end for
    S = {argmin_v dist(v, C_i)} ∪ S.
end for
```

Find the most central vertex in cluster C_i

Seeding Phase



Seeding Phase



Seeding Phase

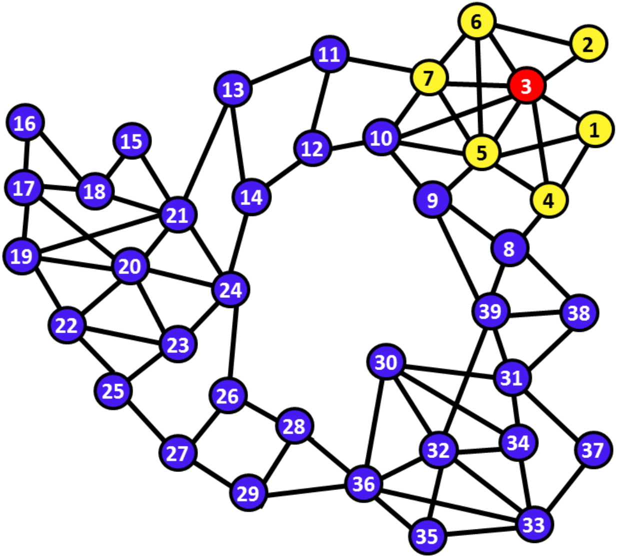
- Spread Hubs
  - Independent set of high-degree vertices

Algorithm 1 Seeding by Spread Hubs

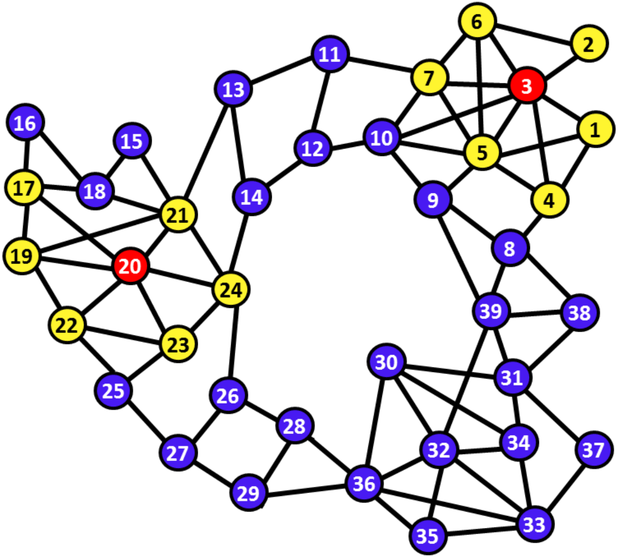
Input: graph $G = (\mathcal{V}, \mathcal{E})$, the number of seeds k.
Output: the seed set $\mathcal{S}$.

Initialize $\mathcal{S} = \emptyset$.
All vertices in $\mathcal{V}$ are unmarked.
while $|\mathcal{S}| < k$ **do**
    Let $\mathcal{T}$ be the set of unmarked vertices with max degree.
    for each $t \in \mathcal{T}$ **do**
        if t is unmarked **then**
            $\mathcal{S} = \{t\} \cup \mathcal{S}$.
            Mark t and its neighbors.
        end if
    end for
end while

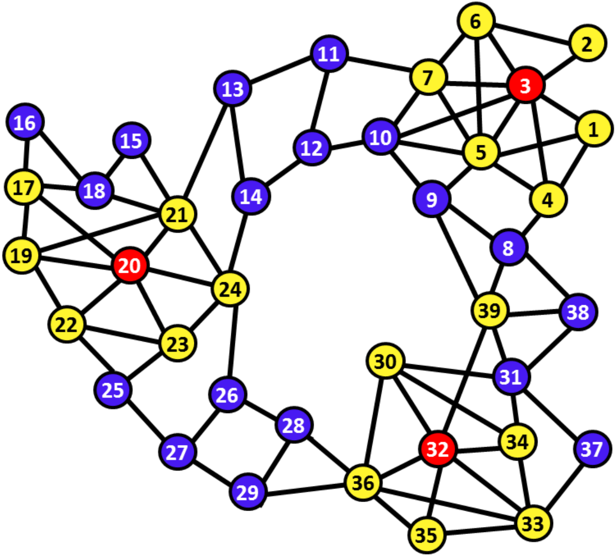
Seeding Phase



Seeding Phase



Seeding Phase



Seeding Phase

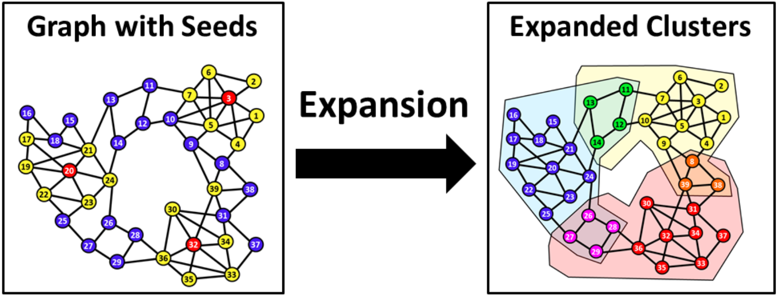
- Other seeding strategies
  - **Local Optimal Egonets.** (Gleich and Seshadhri 2012)
    - ego(s): the egonet of vertex s.
    - Select a seed s such that

$$\texttt{conductance}(\texttt{ego}(s)) \leq \texttt{conductance}(\texttt{ego}(v))$$

      for all v adjacent to s.

  - **Random Seeds.** (Andersen and Lang 2006)
    - Randomly select k seeds.

Seed Set Expansion Phase

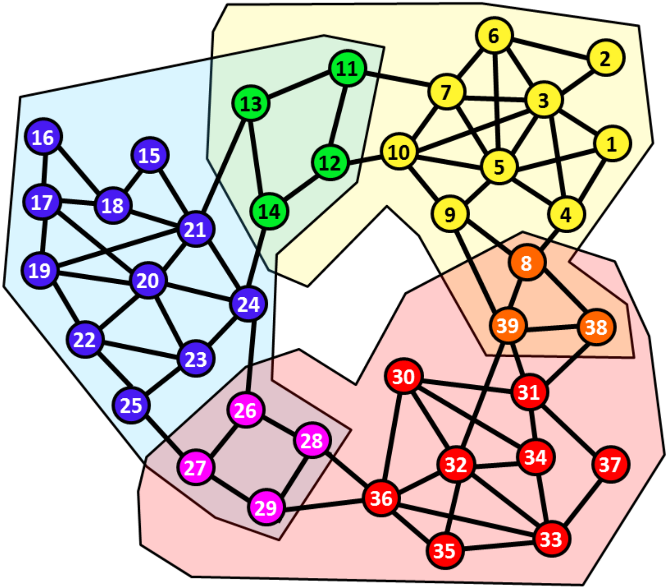


Seed Set Expansion Phase

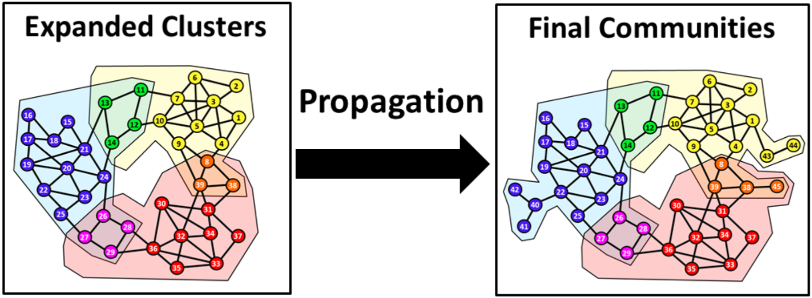
- Personalized PageRank clustering scheme (Andersen et al. 2006)
  1. Given a seed node, compute an approximation of the stationary distribution of a random walk.
  2. Divide the stationary distribution scores by the degree of each node (technical detail needed to remove bias towards high-degree nodes).
  3. Sort the vector, and examine nodes in order of highest to lowest score and compute the conductance score for each threshold cut.

  - Returns a good conductance cluster
  - Remarkably efficient when combined with appropriate data structures
  - For each seed, we use the entire vertex neighborhood as the restart for the personalized PageRank routine.

Seed Set Expansion Phase



Propagation Phase



Propagation Phase

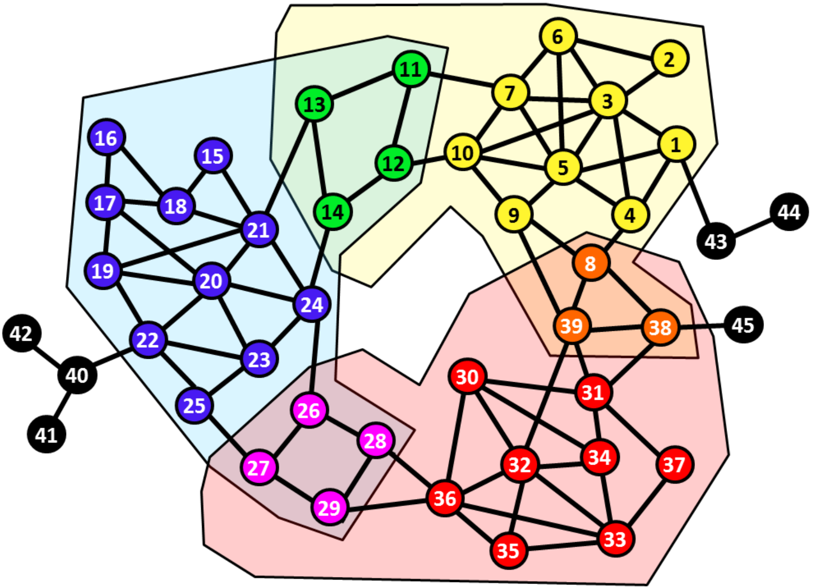
- Each community is further expanded.
- Add whiskers to communities via bridge.

Algorithm 2 Propagation Module

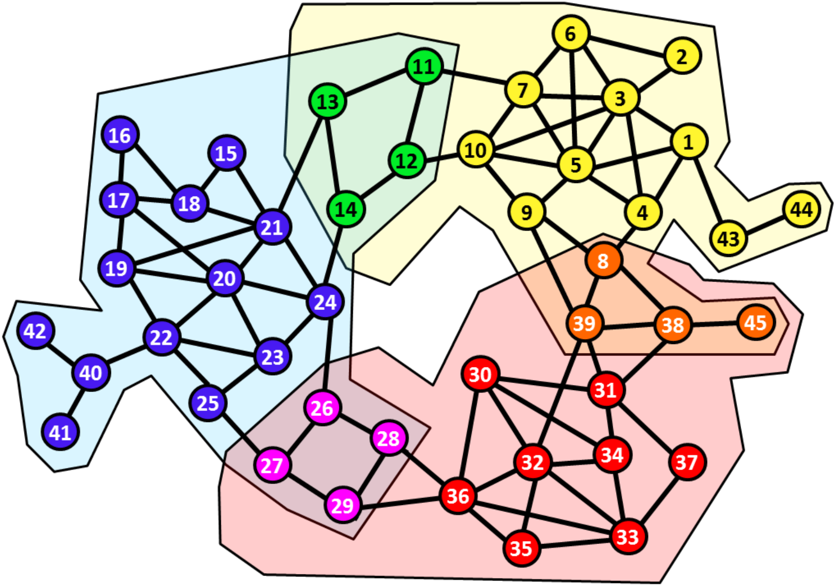
Input: graph $G = (\mathcal{V}, \mathcal{E})$, biconnected core $G_C = (\mathcal{V}_C, \mathcal{E}_C)$, communities of $G_C : \mathcal{C}_i$ $(i = 1, ..., k) \in \mathcal{C}$.
Output: communities of G.

for each $\mathcal{C}_i \in \mathcal{C}$ **do**
    Detect bridges $\mathcal{E}_{B_i}$ attached to $\mathcal{C}_i$.
    for each $b_j \in \mathcal{E}_{B_i}$ **do**
        Detect the whisker $w_j = (\mathcal{V}_j, \mathcal{E}_j)$ which is attached to b_j.
        $\mathcal{C}_i = \mathcal{C}_i \cup \mathcal{V}_j$.
    end for
end for

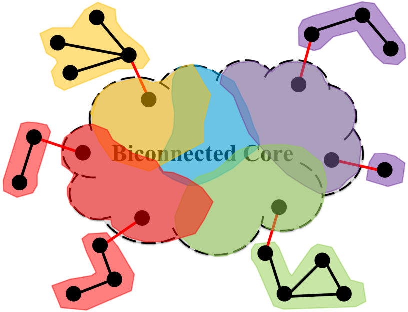
Propagation Phase



Propagation Phase



Propagation Phase



- This process does not increase the cut of each cluster.
- Normalized cut of the expanded cluster is always smaller than or equal to that of original cluster.

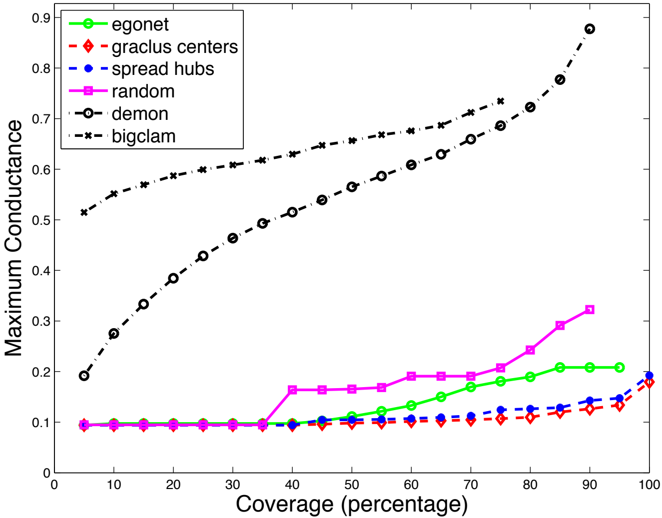
Experimental Results

Experiments

- Comparison with other state-of-the-art methods
  - **Demon** (Coscia et al. 2012)
    - Extracts and computes clustering of ego networks
  - **Bigclam** (Yang and Leskovec 2013)
    - Low-rank non-negative matrix factorization based modeling
- Seed set expansion methods with different seeding strategies
  - **Graclus centers**
  - **Spread hubs**
  - **Local Optimal Egonets** (Gleich and Seshadhri 2012)
  - **Random Seeds** (Andersen and Lang 2006)

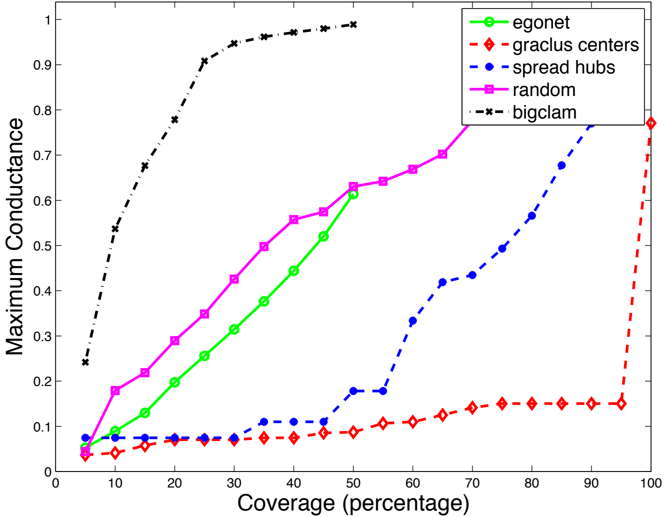
Community Quality using Conductance

- arXiv CondMat collaboration network (21,363 nodes)



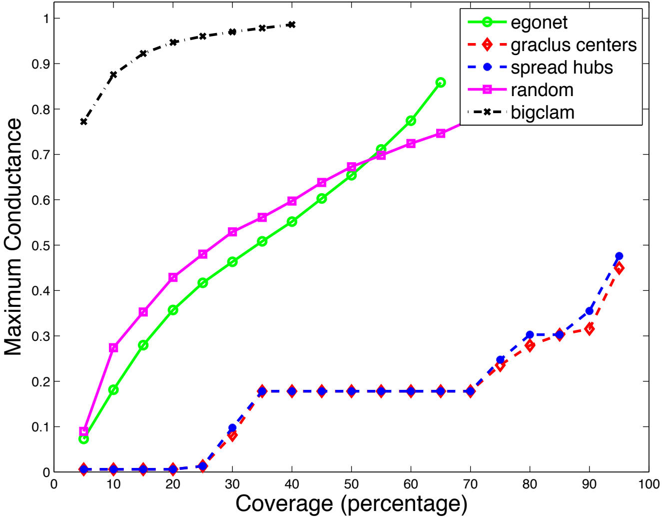
Community Quality using Conductance

- Flickr (1,994,422 nodes)
  - Demon fails on Flickr.



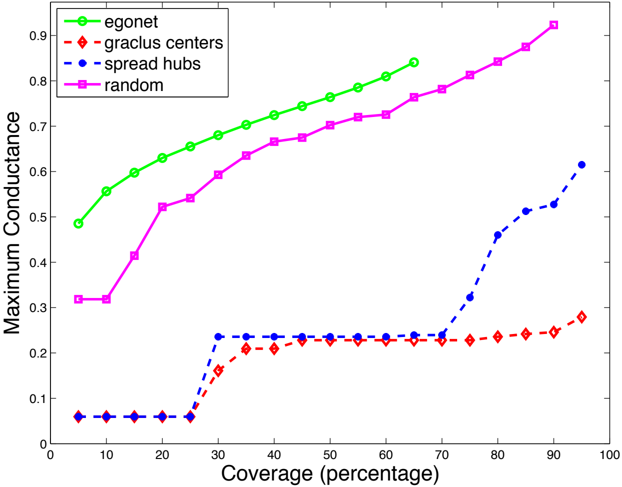
Community Quality using Conductance

- LiveJournal (1,757,326 nodes)
  - Demon fails on LiveJournal.



Community Quality using Conductance

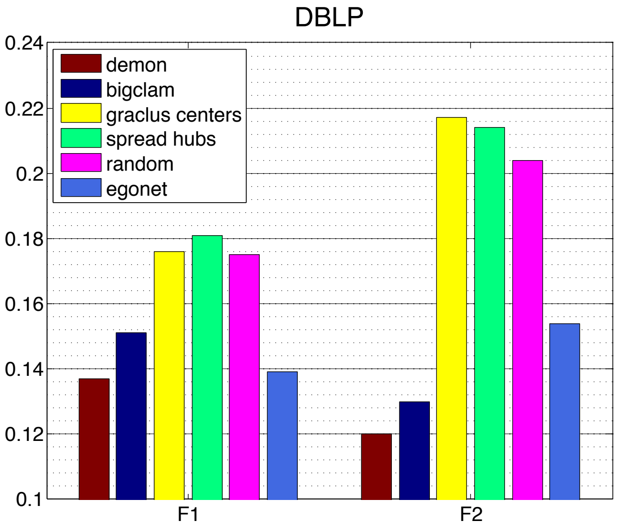
- Myspace (2,086,141 nodes)
  - Demon fails on Myspace.
  - Bigclam does not finish after running for one week.



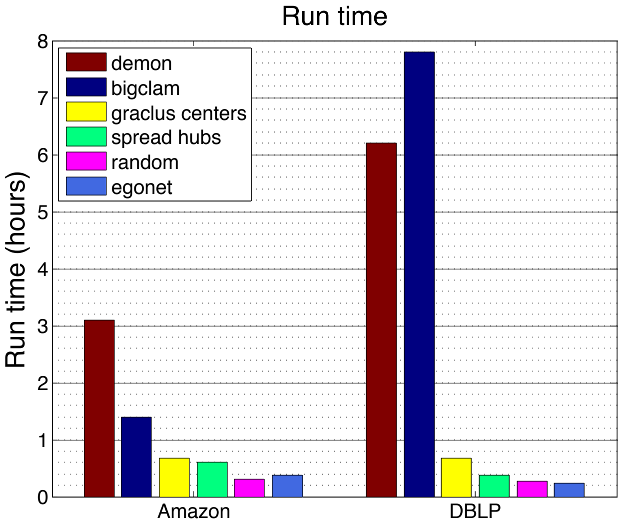
Community Quality via Ground Truth

- Precision
  - how many vertices are actually in the same ground truth community
- Recall
  - how many vertices are predicted to be in the same community in a retrieved community
- Compute F_1, and F_2 measures
  - The ground truth communities are partially annotated.
  - F_2 measure puts more emphasis on recall than precision

Community Quality via Ground Truth



Comparison of Running Times



Conclusions

Conclusions

- Efficient overlapping community detection algorithm
  - Uses a seed set expansion
- Two seed finding strategies
  - Graclus centers
  - Spread hubs
- Our new seeding strategies are better than other strategies, and are thus effective in finding good overlapping clusters in a graph.
- The seed set expansion approach significantly outperforms other state-of-the-art methods.

References

- I. S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors: A multilevel approach. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 29, no. 11, pp. 1944-1957, 2007.
- R. Andersen, F. Chung and K. Lang. Local graph partitioning using PageRank vectors. In FOCS, 2006.
- D. F. Gleich and C. Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In KDD, pages 597-605, 2012.
- R. Andersen and K. J. Lang. Communities from seed sets. In WWW, pages 223-232, 2006.
- J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In WSDM, pages 587-596, 2013.
- M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi. Demon: a local-first discovery method for overlapping communities. In KDD, 2012.