Scalable Graph Clustering using Stochastic Flows
Applications to Community Discovery

Venu Satuluri and Srinivasan Parthasarathy

Data Mining Research Laboratory
Dept. of Computer Science and Engineering
The Ohio State University

http://www.cse.ohio-state.edu/dmrl

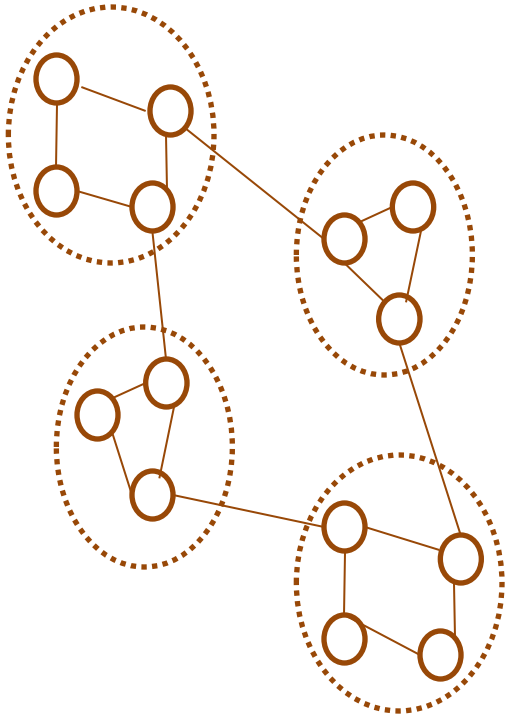
KDD 2009

Outline

- Introduction
  - Problem Statement
  - Markov Clustering (MCL)
- Proposed Algorithms
  - Regularized MCL (R-MCL)
  - Multi-level Regularized MCL (MLR-MCL)
- Evaluation
- Conclusions

Problem Statement

Graph Clustering:
Partition the vertices of a graph into disjoint sets such that each partition is a well-connected/coherent group.



Applications:
- Discovery of protein complexes [Snel '02]
- Community discovery in social networks [Newman '06]
- Image segmentation [Shi '00]

Existing solutions:
- Spectral methods [Shi '00]
- Edge-based agglomerative/divisive methods [Newman '04]
- Kernel K-Means [Dhillon '07]
- Metis [Karypis '98]
- Markov Clustering (MCL) [van Dongen '00]

Markov Clustering (MCL) [van Dongen '00]

The original algorithm for clustering graphs using stochastic flows.

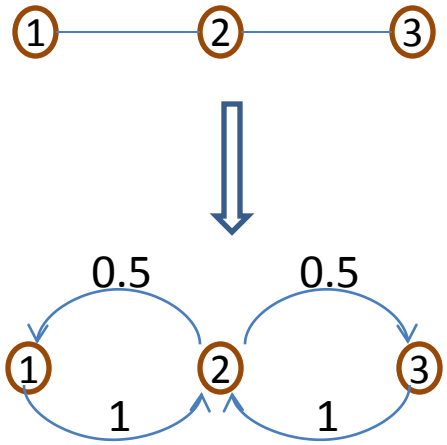
Advantages:
- Simple and elegant.
- Widely used in Bioinformatics because of its noise tolerance and effectiveness.

Disadvantages:
- Very slow.
  - Takes 1.2 hours to cluster a 76K node social network.
- Prone to output too many clusters.
  - Produces 1416 clusters on a 4741 node PPI network.

Can we redress the disadvantages of MCL while retaining its advantages?

Terminology

- **Flow**: Transition probability from a node to another node.
- **Flow matrix**: Matrix with the flows among all nodes; i^{th} column represents flows out of i^{th} node. Each column sums to 1.

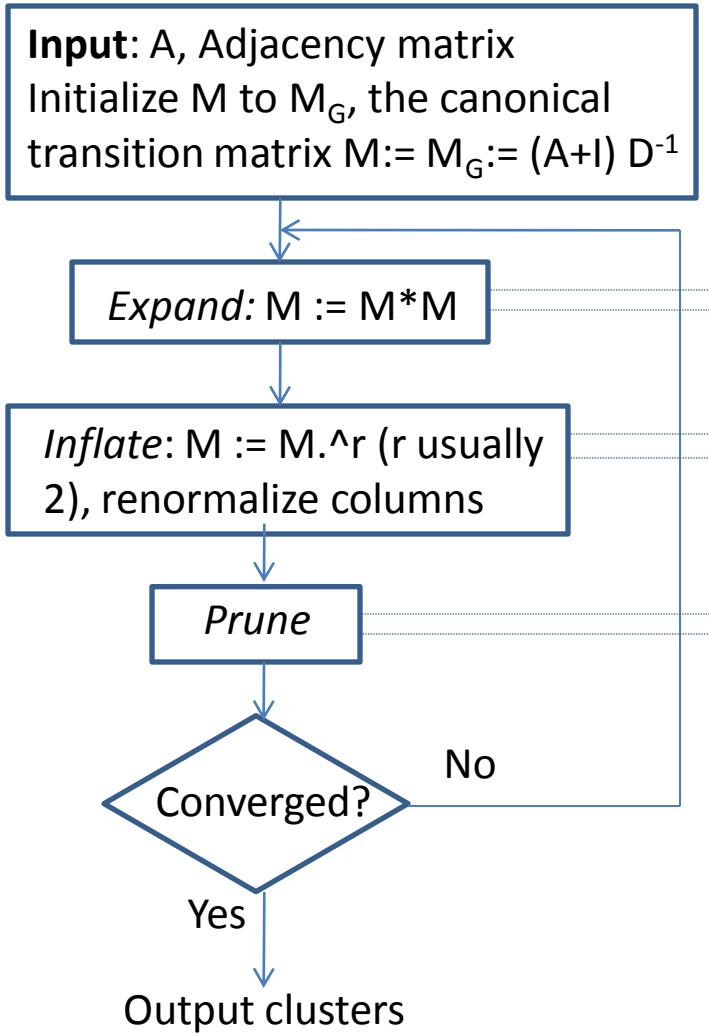


Flow

Matrix

	1	2	3
1	0	0.5	0
2	1.0	0	1.0
3	0	0.5	0

The MCL algorithm



Enhances flow to well-connected nodes as well as to new nodes.

Increases inequality in each column. "Rich get richer, poor get poorer."

Saves memory by removing entries close to zero.

The *Regularize* operator

Why does MCL output many clusters?
Due to overfitting; it does not penalize divergence of flows between neighbors.

Remedy: Penalize divergence in flows between neighbors. Minimize penalty at each node.

$$M'(:,i) = \text{argmin} \sum_{(i,j)\varepsilon E} M_G(j,i) * D(M(:,i)||M(:,j))$$

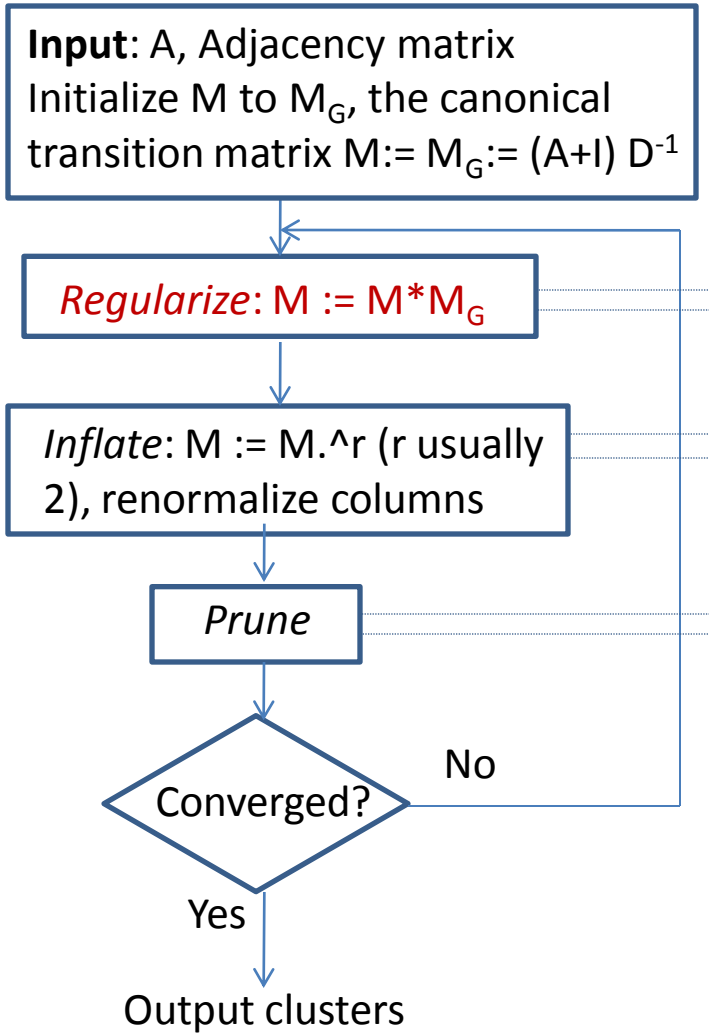
KL Divergence between i and j.

Closed form solution: $M'(:,i) = \sum_{(i,j)\varepsilon E} M_G(j,i)M(:,j)$

This update defines the *Regularize* operator. In matrix notation,

$$Regularize(M) := M*M_G$$
$$= M*(A+I)D^{-1}$$

The Regularized-MCL algorithm

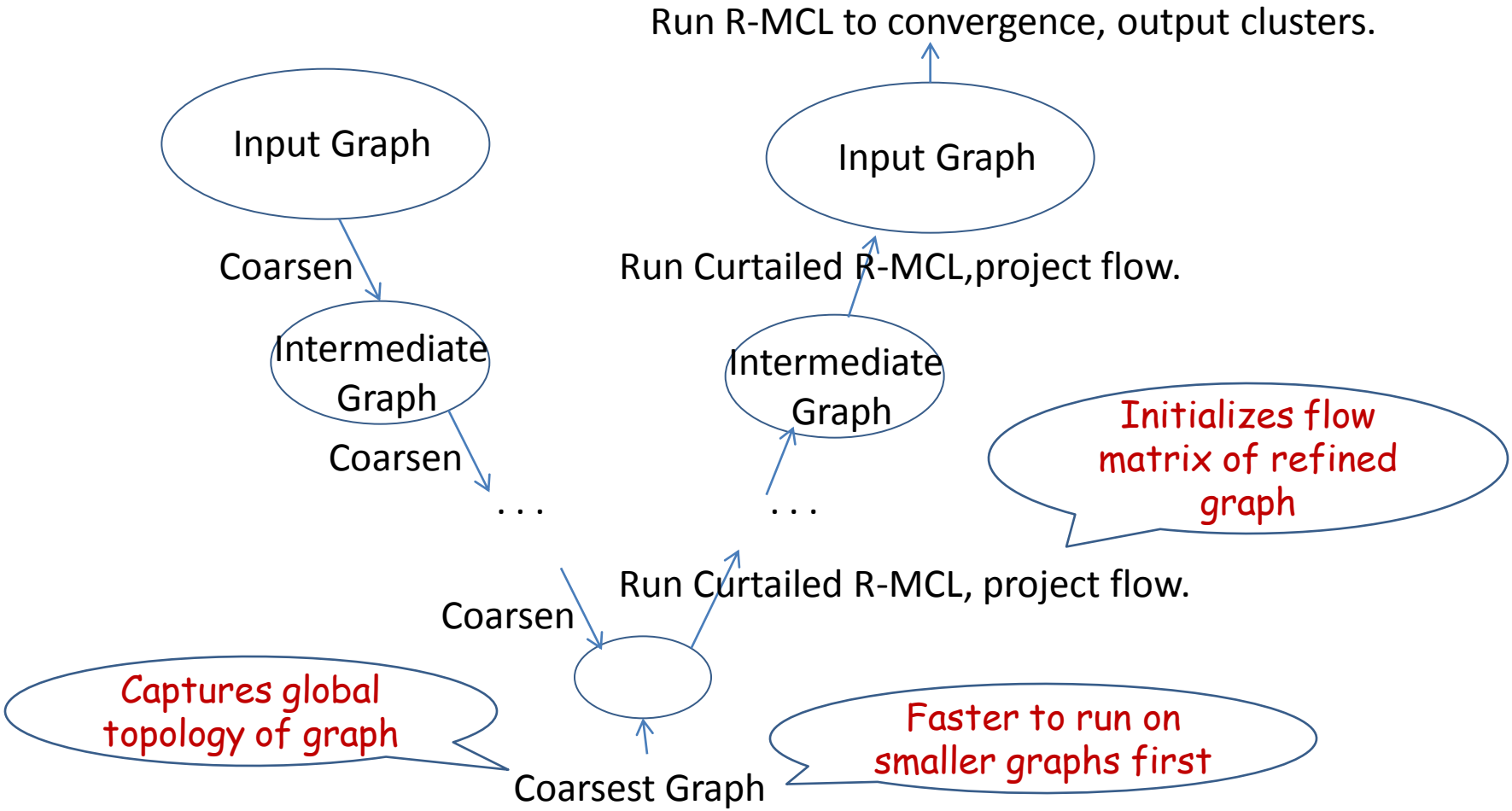


Takes into account flows of the neighbors.

Increases inequality in each column. "Rich get richer, poor get poorer."

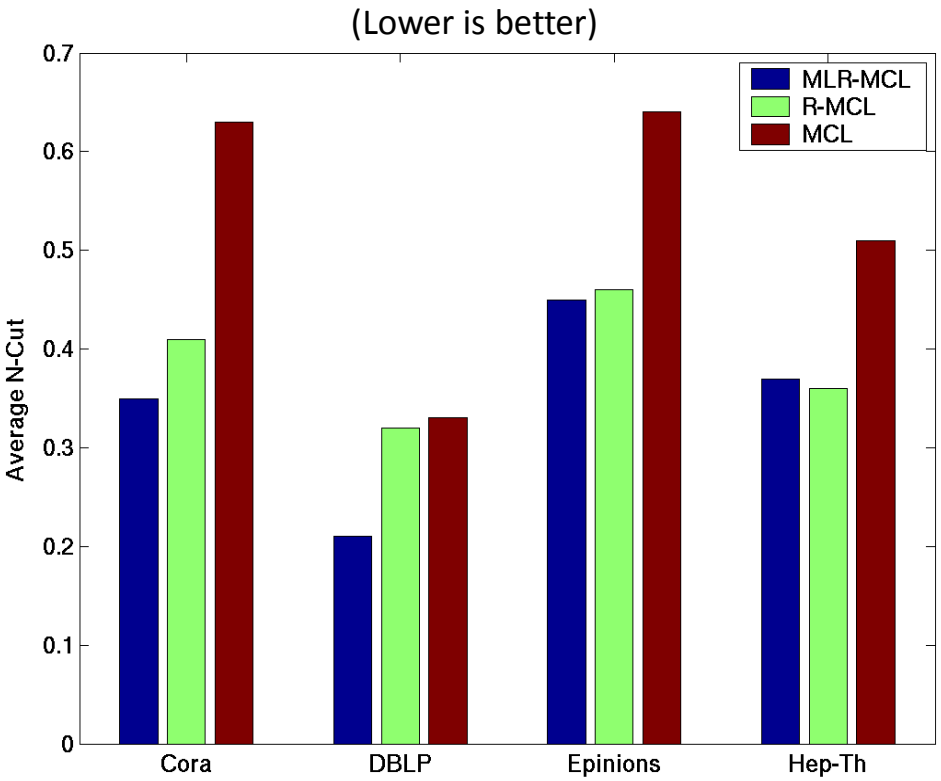
Saves memory by removing entries close to zero.

Multi-level Regularized MCL



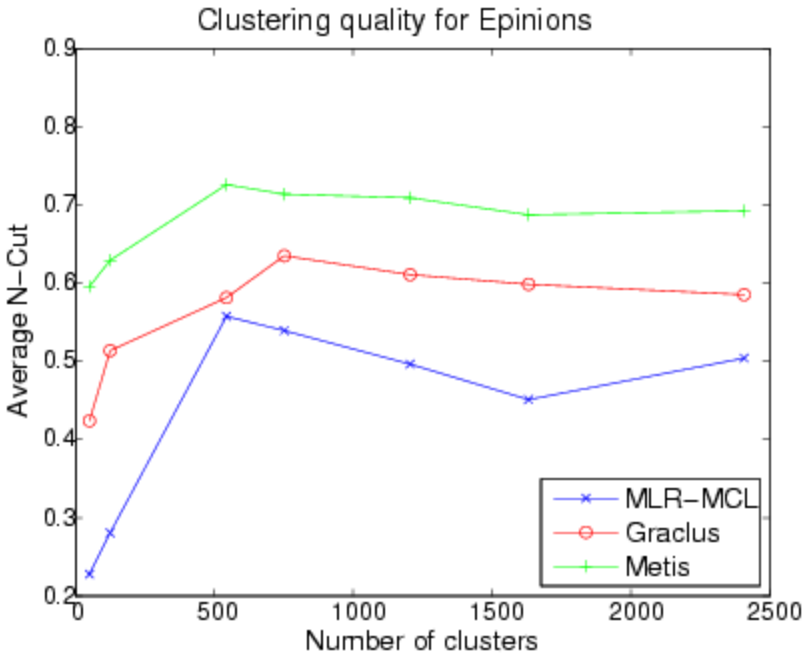
Comparison with MCL

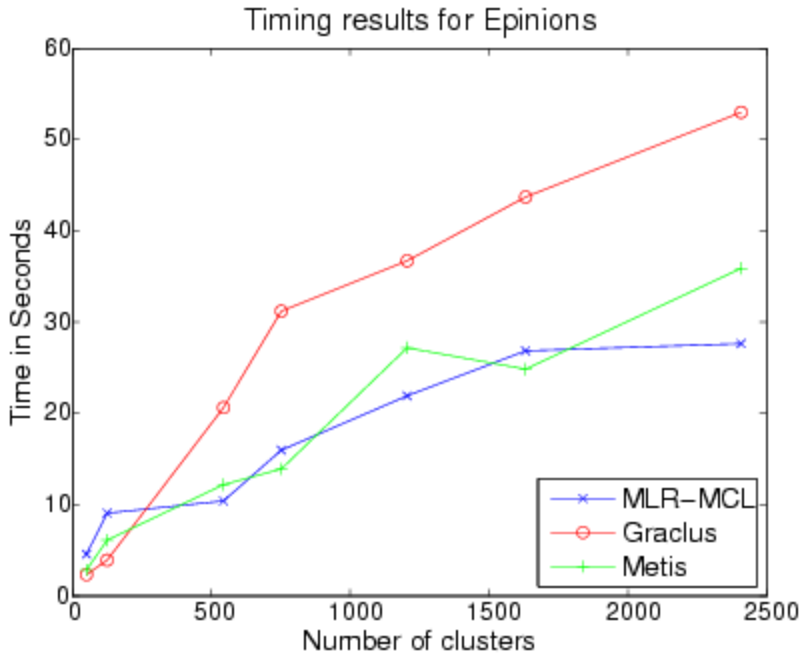
- All three methods run with inflation parameter r=2.
- R-MCL and MLR-MCL output fewer, and better clusters.
- MLR-MCL is on average 96 times faster.
- On the 76K node Epinions graph, MLR-MCL's run time is 26 secs compared to MCL's 1.2 hrs.



MLR-MCL is much faster than MCL, and outputs higher quality clusters.

Comparison with Graclus and Metis



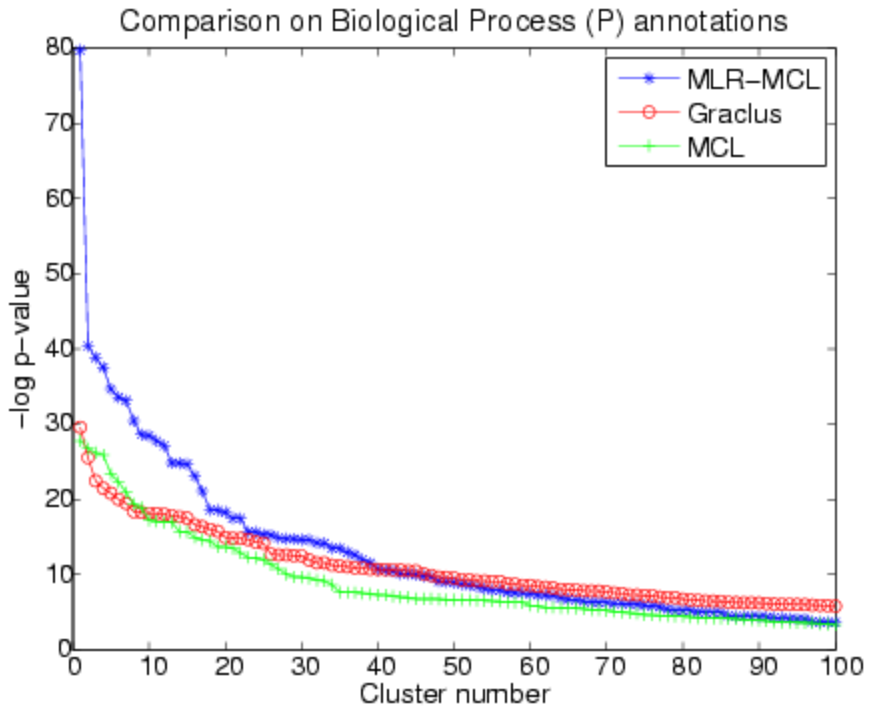


Quality: MLR-MCL improves upon both Graclus and Metis

Speed: MLR-MCL is faster than Graclus and competitive with Metis

Evaluation on PPI networks

Yeast PPI network with 4741 proteins and 15148 interactions.
Annotations from the Gene Ontology database used as ground truth.



MLR-MCL returns clusters of higher biological significance than MCL or Graclus.

Conclusions

- Regularized MCL overcomes the fragmentation problem of MCL.
- Multi-level Regularized MCL further improves quality and speed of R-MCL.
- MLR-MCL often outperforms state-of-the-art algorithms, both quality and speed-wise, on a wide variety of real datasets.

Future Directions:

- Novel coarsening strategies
- Extensions to directed and bi-partite graphs.

Acknowledgements:
This work is supported in part by the following grants: NSF CAREER IIS-0347662, RI-CNS-0403342, CCF-0702586 and IIS-0742999

References:

1. MCL - *Graph Clustering by Flow Simulation.* S. van Dongen, Ph.D. thesis, University of Utrecht, 2000.
2. Graclus - *Weighted Graph Cuts without Eigenvectors: A Multilevel Approach.* Dhillon et. al., IEEE. Trans. PAMI, 2007.
3. Metis - *A fast and high quality multilevel scheme for partitioning irregular graphs.* Karypis and Kumar, SIAM J. on Scientific Computing, 1998
4. *Normalized Cuts and Image Segmentation*. Shi and Malik, IEEE. Trans. PAMI, 2000.
5. *Finding and evaluating community structure in networks*. Newman and Girvan, Phys. Rev. E 69, 2004.
6. *The identification of functional modules from the genomic association of genes.* Snel et. al., PNAS 2002.

Thank You!