Newman & Girvan

COMMUNITY DETECTION AND CLUSTERING IN GRAPHS

Vaibhav Mallya

EECS 767

D. Radev

AGENDA

- Agenda
- Basic Definitions
- Girvan-Newman Algorithm
- Donetti-Munoz Spectral Method
- Karypis-Kumar Multi-level Partitioning
- Graclus
- GraphClust

DEFINITIONS

- Complex networks: Study of representations of interactions between real-world entities
- Borrows heavily from classical graph theory
- Vertices, nodes, entities connected by links, edges, connections

DEFINITIONS (CONT.)

- Community: One of many (possibly overlapping) subgraphs
  - Has strong internal node-node connections
  - Weaker external connections
  - Community detection algorithms stress high internal connectivity and low external connectivity with a given community
- Cluster: See community

DEFINITIONS (CONT.)

- Practical applications of clustering, community detection
  - Simplifies visualization, analysis on complex graphs
  - Search engines – Categorization
    - Clusty.com was an early one, is still around
  - Social networks - Useful for tracking group dynamics
  - Neural networks - Tracks functional units
  - Food webs - helps isolate co-dependent groups of organisms

GIRVAN-NEWMAN ALGORITHM

- Published by Michelle Girvan and Mark Newman in 2002
- Helped rekindle recent interest in community detection
- Iteratively finds community boundaries
- Hierarchical divisive algorithm

GIRVAN-NEWMAN ALGORITHM

- Calculate edge-betweenness for all edges
- Remove the edge with highest betweenness
- Recalculate betweenness
- Repeat until all edges are removed, or modularity function is optimized (depending on variation)

GIRVAN-NEWMAN ALGORITHM

- Quality Function: Objective measure of how good community divisions are
- Girvan-Newman Modularity is most popular quality function
  - Premise: Random graph has no communities
  - Create a null model of the original graph – similar to actual graph, but without communities.
  - May involve, f.ex randomly rewiring certain nodes while maintaining their degree
  - Hereafter referred to simply as "modularity"

GIRVAN-NEWMAN ALGORITHM

- Modularity equation

$$Q = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) s_i s_j = \frac{1}{4m} \mathbf{s}^T \mathbf{B} \mathbf{s}.$$

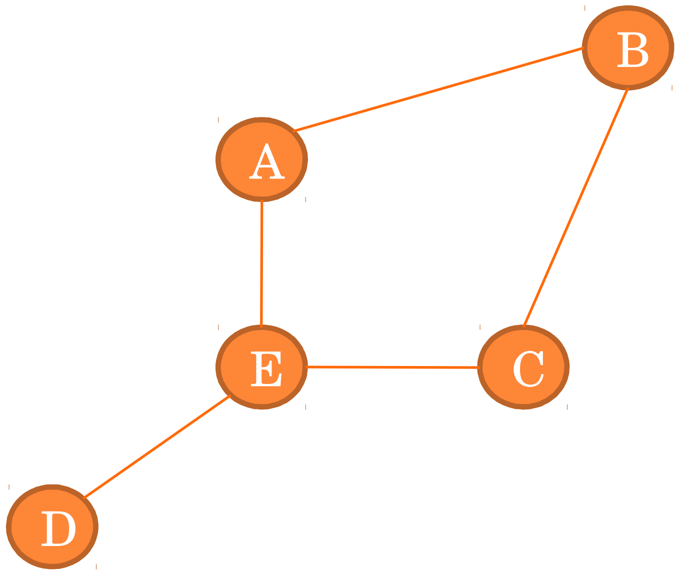
- (Obviously, we can try and optimize this function directly; there are other approaches that do this)

GIRVAN-NEWMAN ALGORITHM

- Vertex Betweenness Centrality
  - First proposed by Freeman
  - Classical vertex importance measure on a network
  - Defined as the total number of shortest paths that pass through each vertex on the network
  - There is a possible ambiguity with this definition

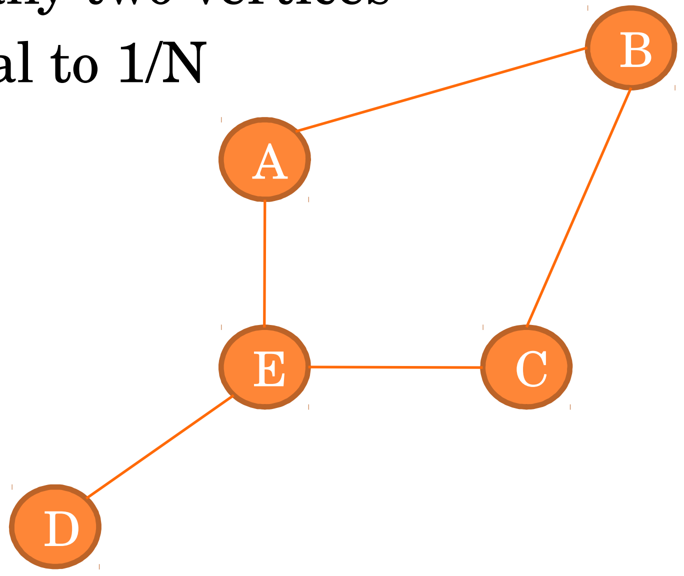
GIRVAN-NEWMAN ALGORITHM

- See it?



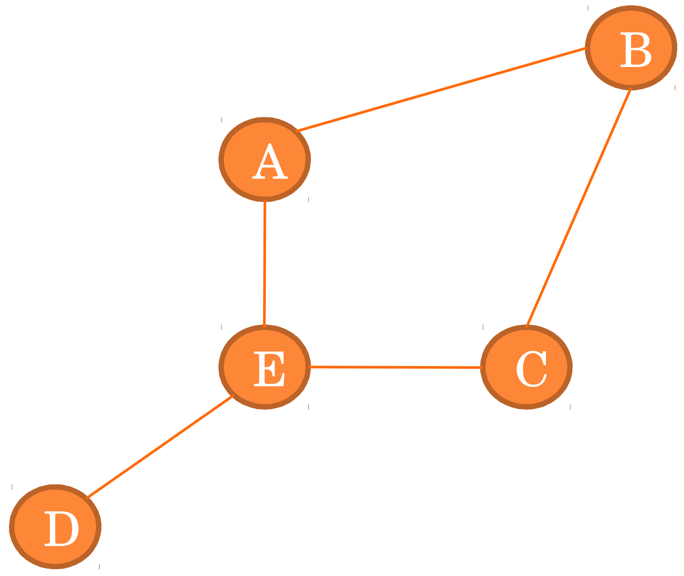
GIRVAN-NEWMAN ALGORITHM

- To resolve:
  - We're actually calculating all-shortest paths between two paths
  - There are N paths between any two vertices
  - Each path gets a weight equal to 1/N



GIRVAN-NEWMAN ALGORITHM

- Vertex betweenness of C:
    - D-B +0.5
    - E-B +0.5
    - C-B +1
    - A-B +1
    - = 2



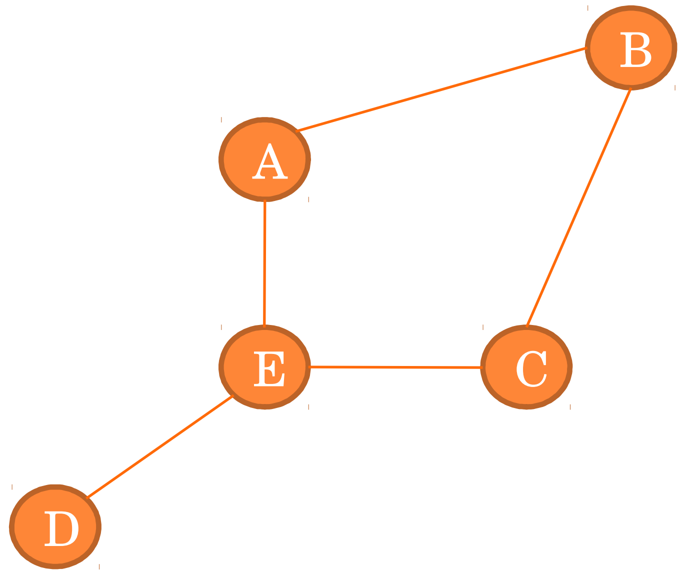
GIRVAN-NEWMAN ALGORITHM

- Edge betweenness centrality
  - G-N's generalization of vertex betweenness
  - Number of shortest paths that pass through a given edge
  - "If there is more than one shortest path between a pair of vertices, each path is given equal weight such that the total weight of all the paths is unity"

GIRVAN-NEWMAN ALGORITHM

- Edge Betweenness example - EA

D-B	+0.5
E-B	+0.5
E-A	+0.5
=	1.5



GIRVAN-NEWMAN ALGORITHM

- Algorithm for all-edge betweenness
- Choose two vertices
  - Calculate all shortest paths between these two vertices
  - Place every path encountered into a set
  - Iterate over the set
  - Increment the betweenness of every path by 1 / size of the set
- Repeat for every pair of vertices

GIRVAN-NEWMAN ALGORITHM

- Intuitively, why should this work? Analogy:
  - Network of N nodes: nodes are towns, edges are roads
  - Place N-1 cars on each node; each one to a town
  - Each road gets a point when a car drives on it
  - Remove the highest ranked road – interstate highway
  - Repeat the process
  - First we'll remove all interstates (leaving state roads)
  - Then state roads will be removed, leaving county roads, then suburban roads, etc
  - After we each set of levels, we get a more fine-grained division of communities

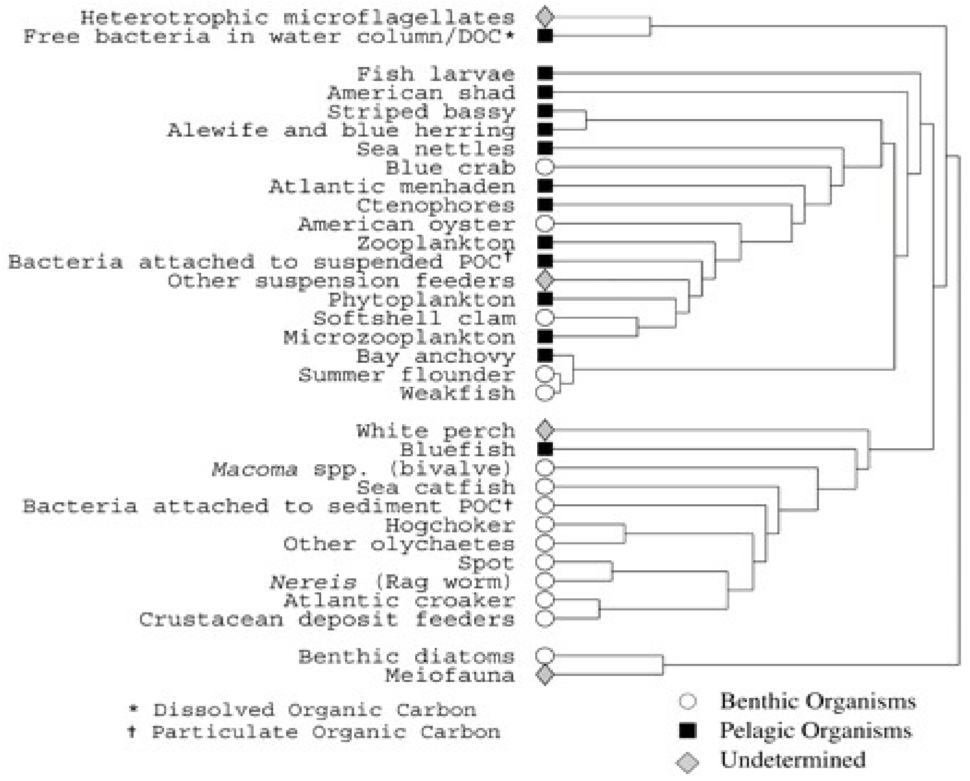
GIRVAN-NEWMAN COMPLEXITY

- Original definition for undirected graphs
  - More on possible directed generalizations later

- O(mn), or O(n^2) on sparse graph

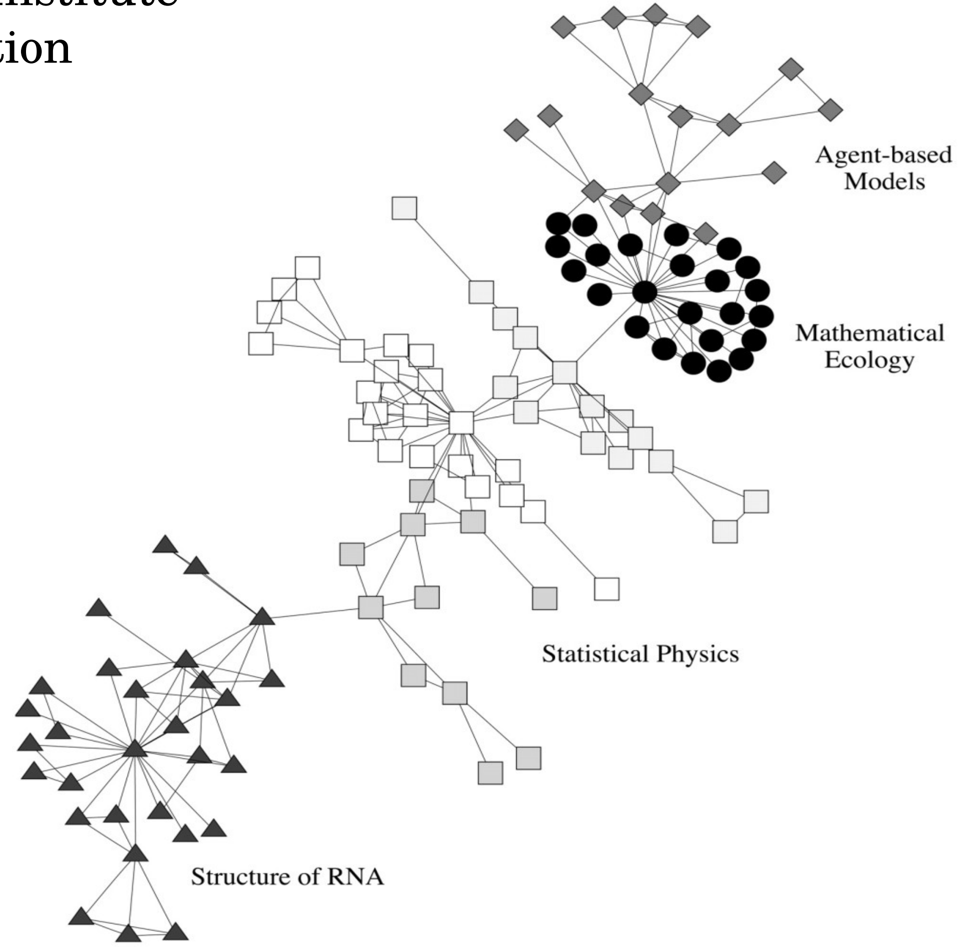
GIRVAN-NEWMAN OUTPUT

Chesapeake
Bay Food Web



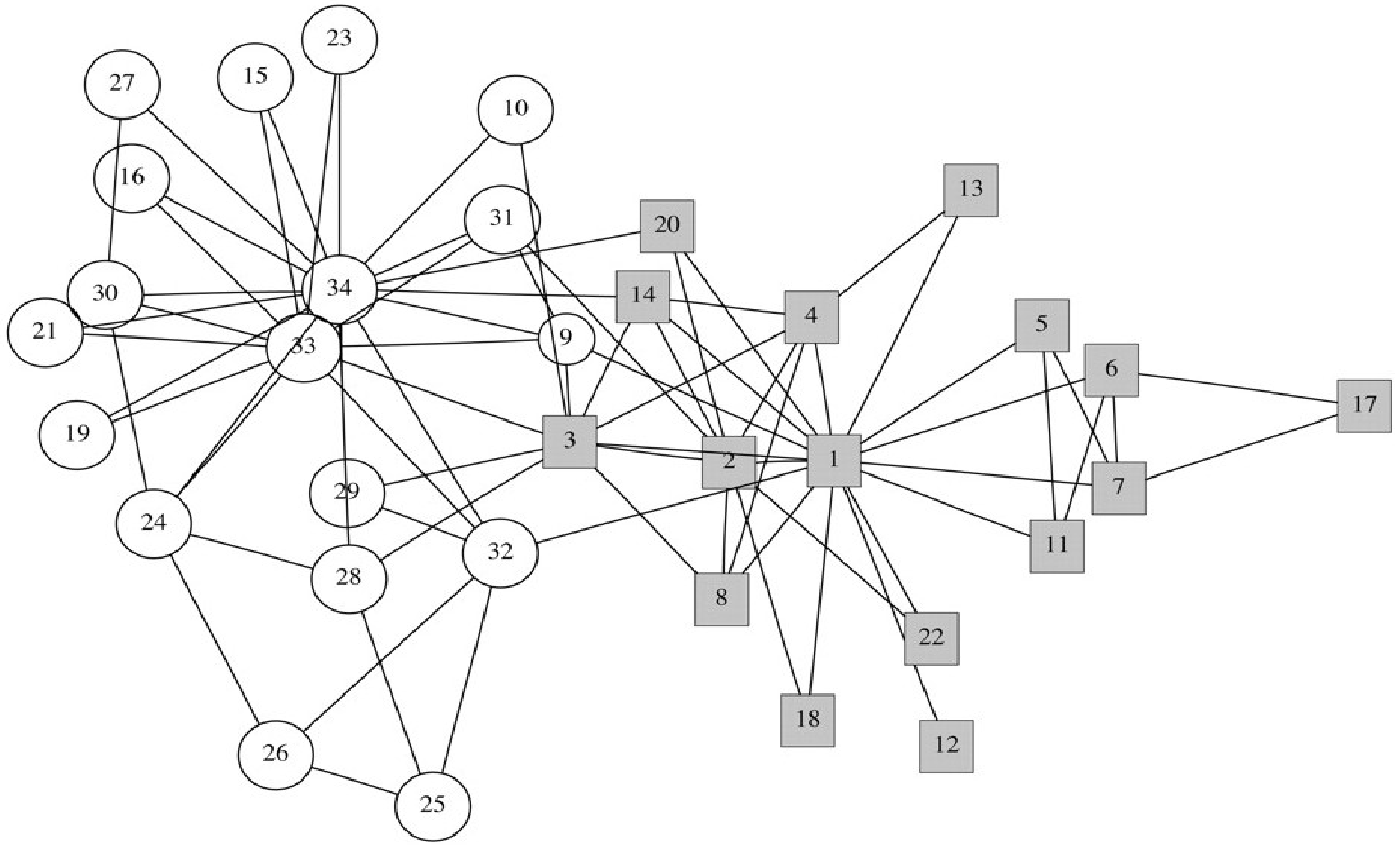
GIRVAN-NEWMAN OUTPUT

Santa Fe Institute
Collaboration
Network



GIRVAN-NEWMAN OUTPUT

- Zachary's Karate Club



DONETTI-MUNOZ SPECTRAL METHOD

- Based on eigenvectors of the Laplacian matrix
- “Elegant insight”
  - For any two nodes A and B, these eigenvector components will be very close
  - Convert into coordinates: points in an M-dimensional metric space
  - M corresponds to number of eigenvectors used
- Then we can apply standard clustering techniques
  - Manhattan distance, angle distance, etc
- Apply basic hierarchical clustering methods after conversion; merge connected cluster pairs

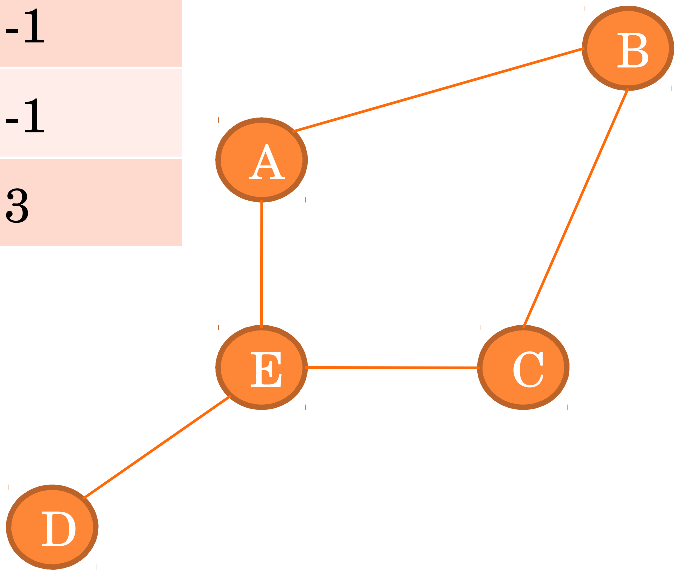
DONETTI-MUNOZ SPECTRAL METHOD

- Laplacian Matrix (aka Kirchhoff matrix) “L”
  - Encodes topological information
  - Symmetric N x N
  - Each diagonal “i” corresponds to degree of vertex “i“
  - Other elements = -1 edge exists between row and column, 0 otherwise
  - Useful properties

DONETTI-MUNOZ SPECTRAL METHOD

- Example Laplacian

	A	B	C	D	E
A	2	-1	0	0	-1
B	-1	2	-1	0	0
C	0	-1	2	0	-1
D	0	0	0	1	-1
E	-1	0	-1	-1	3



DONETTI-MUNOZ SPECTRAL METHOD

- Calculation of Laplacian’s eigenvectors is bottleneck

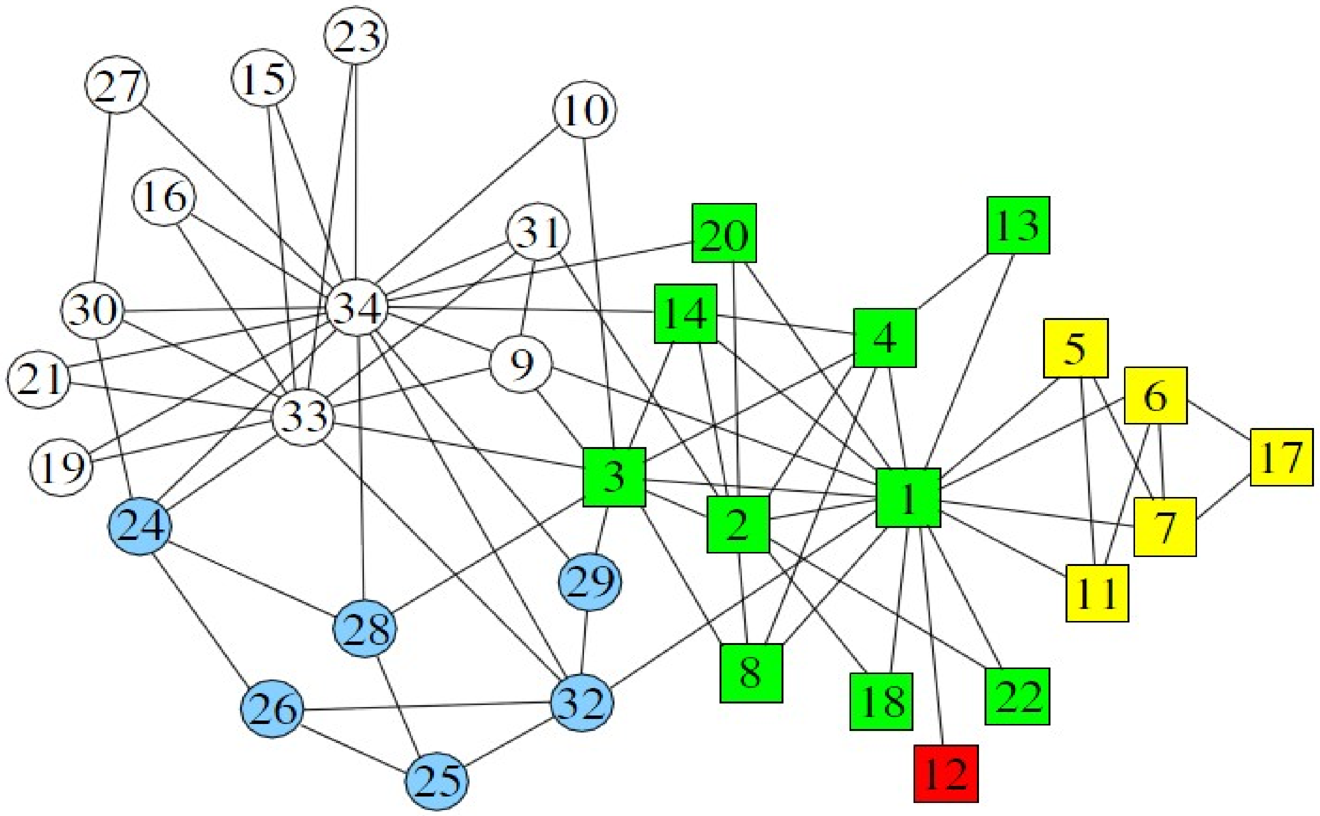
- Use Lanczos method to determine best ones
  - Runtime: O(Number of edges / (difference between two smallest eigenvalues))

DONETTI-MUNOZ SPECTRAL METHOD

- Lanczos Method
  - Iterative algorithm for approximating eigenvalues of a square matrix
  - Improves on power-law method – retain intermediate eigenvalues
  - Generates tri-diagonal matrix whose eigenvectors are calculated cheaply
  - Must balance round-off errors, which accumulate – numerical stability is complicated but manageable

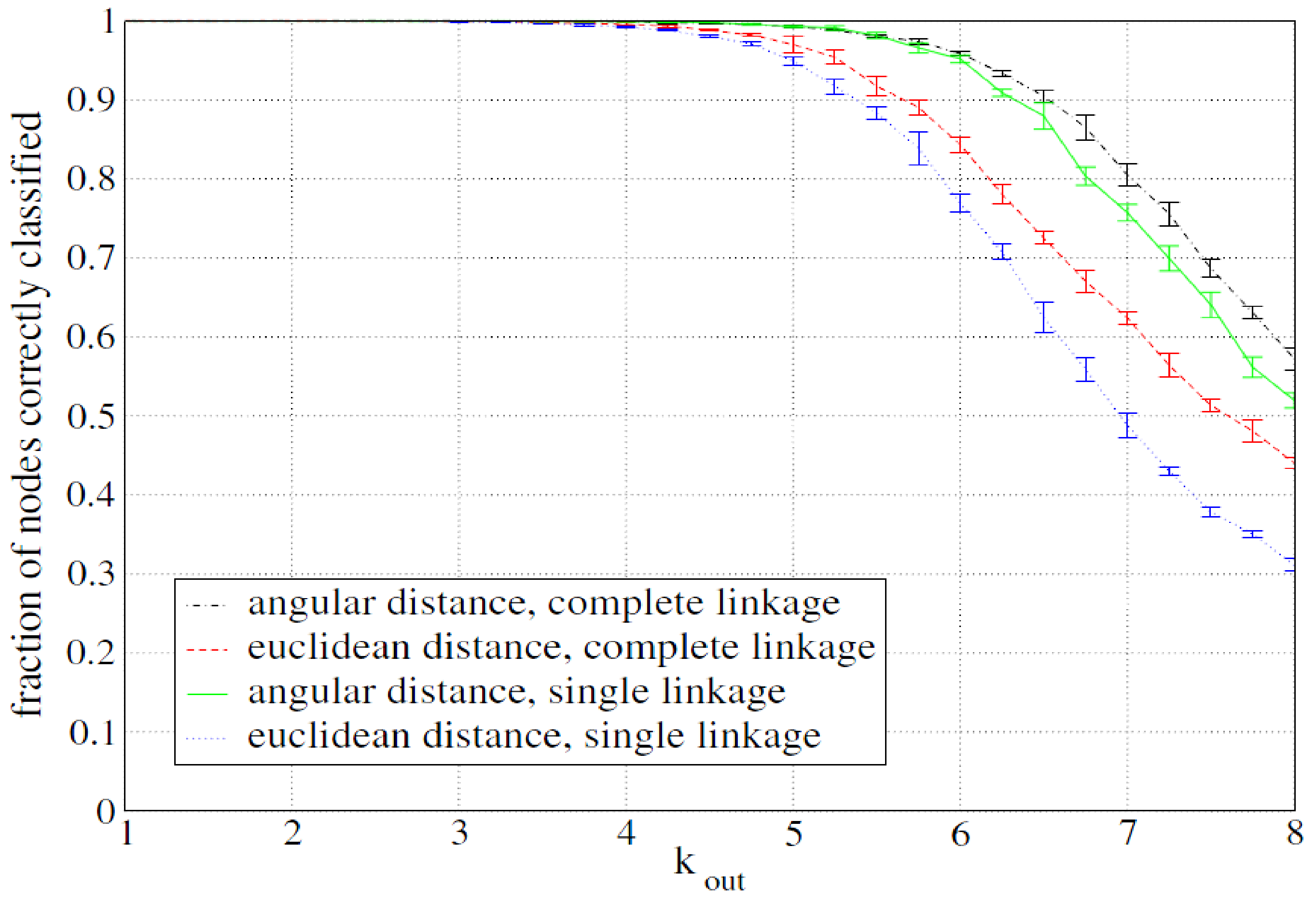
DONETTI-MUNOZ SPECTRAL METHOD

- Applied to Zachary's Karate Club Network
  - Circles/squares = original, colors = additional



DONETTI-MUNOZ SPECTRAL METHOD

- Applied to computer-generated random graphs



KARYPIS-KUMAR MULTI-LEVEL PARTITIONING

- Example of multi-level partitioning scheme
  1) Coarsen: Reduce graph in some meaningful way, collapsing vertices and edges
  2) Partition: Calculate good partition
  3) Uncoarsen: Obtain something resembling the original graph, with partitions intact.

- Key advantage of such schemes is speed
  - Partitions tend to be of slightly worse quality
  - But the algorithm is significantly faster

KARYPIS-KUMAR MULTI-LEVEL PARTITIONING

- Karypis-Kumar based their algorithm on Hendrickson-Leland collapsing approach
  - Heavy Edge Matching for partitioning
  - Improved Kernigan-Lin for partition refining

1) Coarsen: Collapse a pair of adjacent vertices
    1) Creates multinode of the vertices
    2) They use "maximal matchings" to include maximum possible number of collapsed edges
2) Partition: Compute a high-quality bisection
    1) Cut the graph in a meaningful way
    2) KL, SB, GGP, GGGP

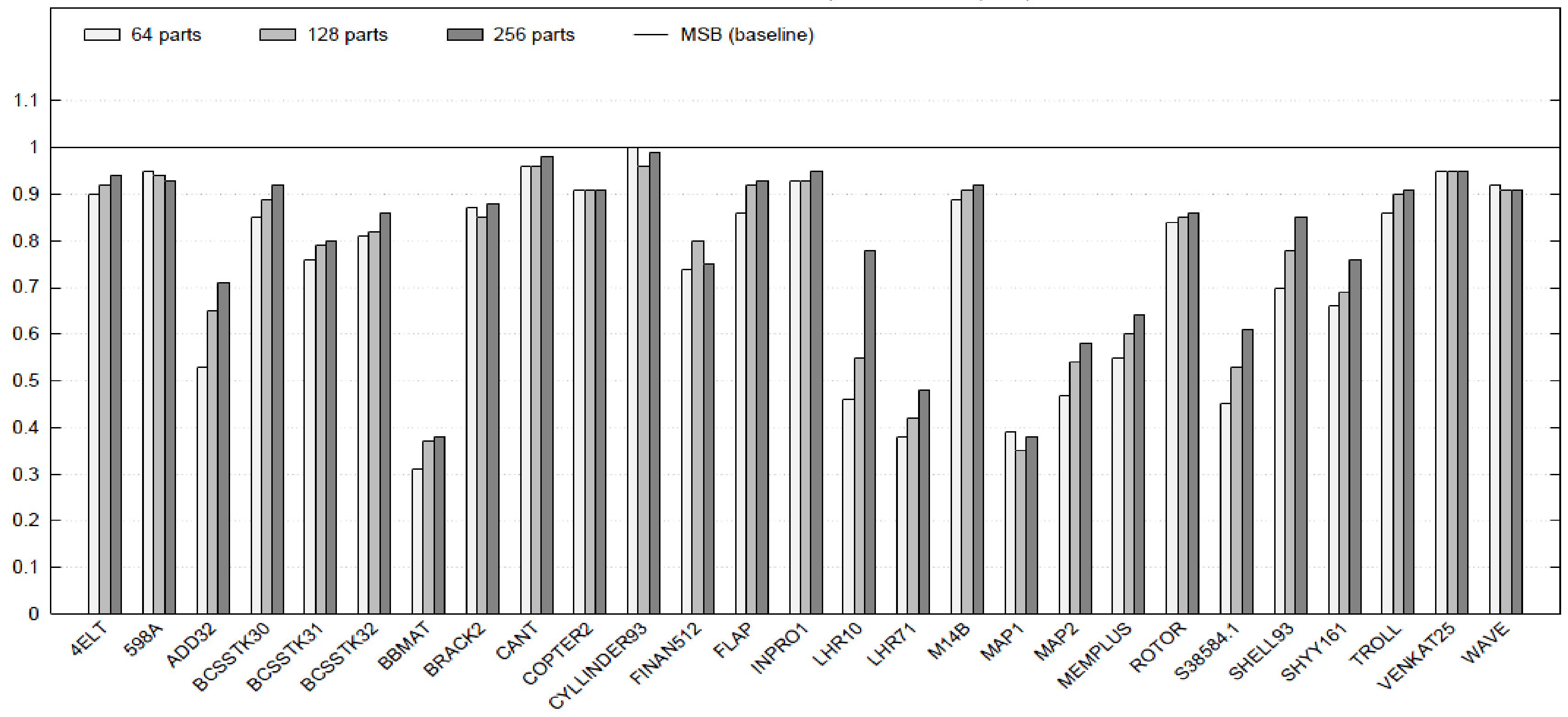
KARYPIS-KUMAR MULTI-LEVEL PARTITIONING

3) Uncoarsening: Expand the smaller graph back into the original via boundary KL method
    1) The graph is uncoarsened in stages
    2) At each stage, the partition is corrected

KARYPIS-KUMAR MULTI-LEVEL PARTITIONING

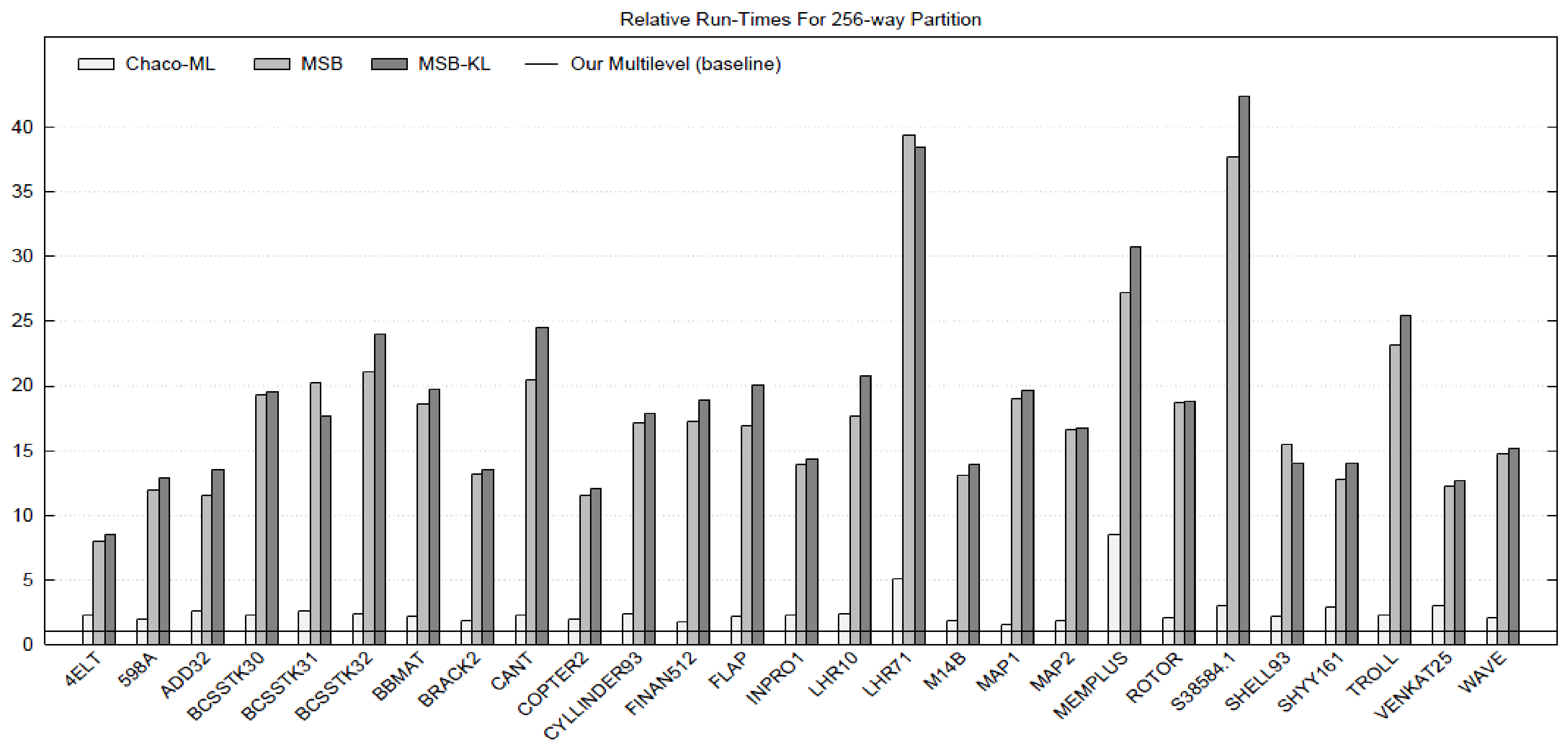
- The graph that shows their algorithm is better
  - Compared to MS Bisection algorithm.

Our Multilevel vs Multilevel Spectral Bisection (MSB)



KARYPIS-KUMAR MULTI-LEVEL PARTITIONING

- Another graph showing their algorithm is better



GRACLUS

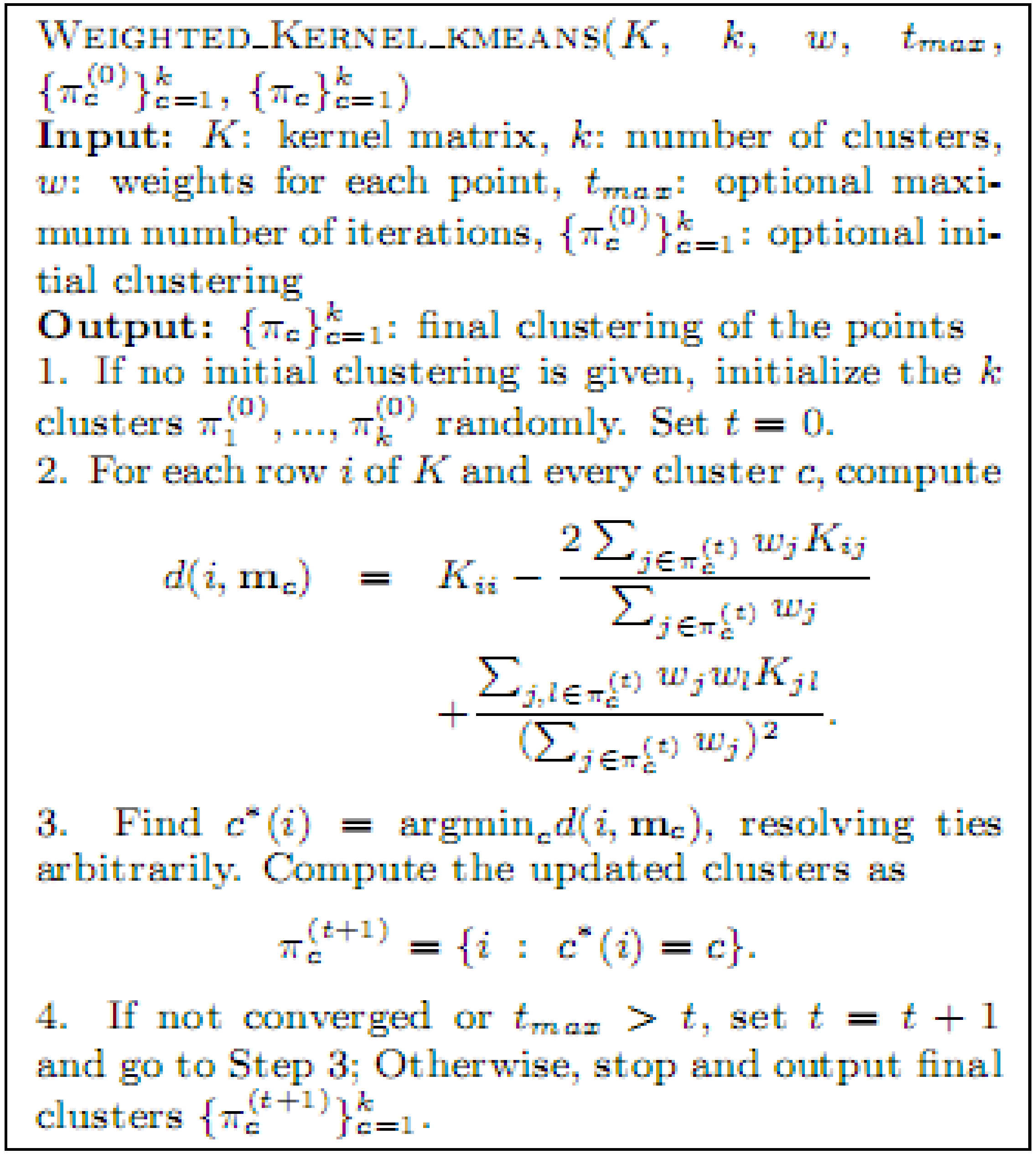
- Software for calculating partitions, clusters
  - Uses novel algorithms developed at UT Austin
- Avoids use of eigenvector-based schemes
  - Eigenvectors are Expensive to compute
- Key idea: Finding graph clusters equivalent to maximizing weighted kernel k-means objective function

GRACLUS

- Like Karypis-Kumar, multi-level
- Coarsening: Yu-Shi algorithm, a kind of spectral method

GRACLUS

- Refinement



WEIGHTED_KERNEL_KMEANS(K, k, w, t_{max}, $\{\pi_c^{(0)}\}_{c=1}^k$, $\{\pi_c\}_{c=1}^k$)

Input: K: kernel matrix, k: number of clusters, w: weights for each point, t_{max}: optional maximum number of iterations, $\{\pi_c^{(0)}\}_{c=1}^k$: optional initial clustering

Output: $\{\pi_c\}_{c=1}^k$: final clustering of the points

1. If no initial clustering is given, initialize the k clusters $\pi_1^{(0)}, \ldots, \pi_k^{(0)}$ randomly. Set $t = 0$.
2. For each row i of K and every cluster c, compute

$$d(i, \mathbf{m}_c) = K_{ii} - \frac{2\sum_{j\in\pi_c^{(t)}} w_j K_{ij}}{\sum_{j\in\pi_c^{(t)}} w_j} + \frac{\sum_{j,l\in\pi_c^{(t)}} w_j w_l K_{jl}}{(\sum_{j\in\pi_c^{(t)}} w_j)^2}.$$

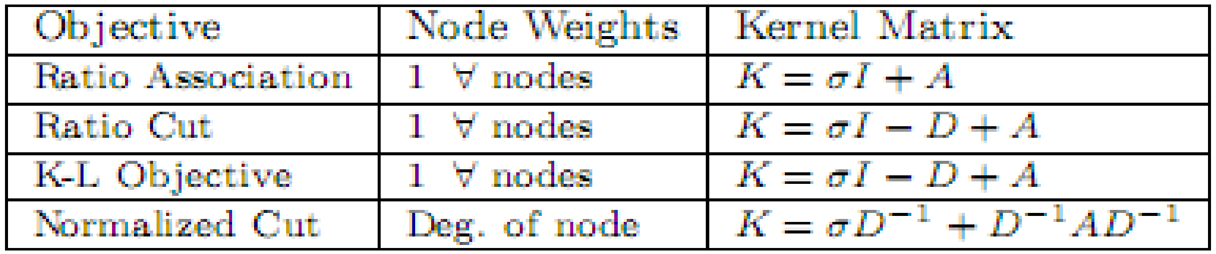
3. Find $c^*(i) = \operatorname{argmin}_c d(i, \mathbf{m}_c)$, resolving ties arbitrarily. Compute the updated clusters as

$$\pi_c^{(t+1)} = \{i : c^*(i) = c\}.$$

4. If not converged or $t_{max} > t$, set $t = t + 1$ and go to Step 3; Otherwise, stop and output final clusters $\{\pi_c^{(t+1)}\}_{c=1}^k$.

GRACLUS

- Refinement – Objectives, Kernel Matrix to use



Objective	Node Weights	Kernel Matrix
Ratio Association	1 $\forall$ nodes	$K = \sigma I + A$
Ratio Cut	1 $\forall$ nodes	$K = \sigma I - D + A$
K-L Objective	1 $\forall$ nodes	$K = \sigma I - D + A$
Normalized Cut	Deg. of node	$K = \sigma D^{-1} + D^{-1} A D^{-1}$

GRACLUS

Figure 2: Quality and computation time of our multilevel methods compared with the benchmark spectral algorithm. The left panel plots the ratio of the ratio association value of every algorithm to that of mlkkm(20) for 64 clusters. Note that bars below the baseline correspond to cases where mlkkm(20) performs better. The right panel compares computation times for generating 64 clusters using different algorithms. As an example, on the ADD32 graph, the spectral algorithm is 48 times slower than mlkkm(20) and 58 times slower than mlkkm(0).

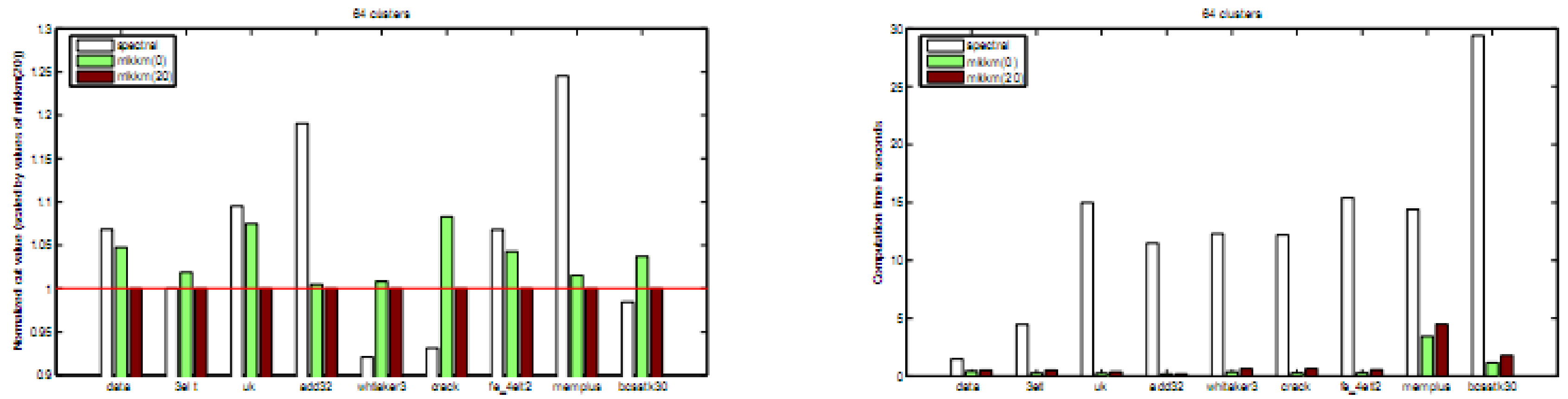
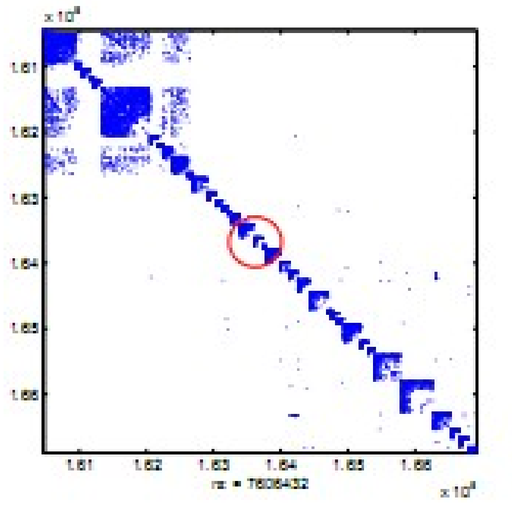
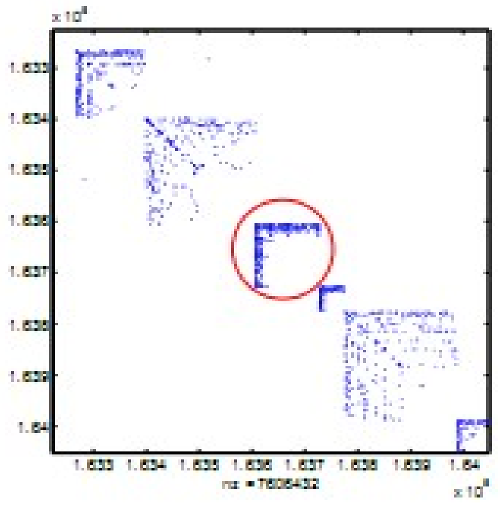


Figure 3: Quality and computation time of our multilevel methods compared with the benchmark spectral algorithm. The left panel plots the ratio of the normalized cut value of every algorithm to that of mlkkm(20) for 64 clusters. Note that bars above the baseline correspond to the cases where mlkkm(20) performs better. The right panel compares computation times for generating 64 clusters. As an example, on the ADD32 graph, the spectral algorithm is 58 times slower than mlkkm(20) and 60 times slower than mlkkm(0).

GRACLUS

- Harry Potter





GRAPHCLUST

- Shasha et al, primarily from NYU
- “Clusters graphs based on topology”
- Clusters a *set* of graphs
- Clusters a graph based on four dimensions:
  - Substructure definition
  - Graph type
  - Distance Metric
  - Number of Clusters
- 16 algorithms; each dimension has 1 option.

GRAPHCLUST

- For each graph
  - Record the number of times each substructure is present
  - Construct a vector of non-negative integers
  - Use the provided distance metric (inner dot product or Euclidean distance) to cluster the graph.
- Intended to work on *database* of graphs, potentially thousands of them or more
- Algorithm is supposed to be fast
  - Can optionally use SVD for added speed

LIST OF SOURCES

- A fast and high quality multilevel scheme for partitioning irregular graphs G. Karypis and V. Kumar, 1998
- Community structure in social and biological networks (Girvan-Newman, 2002) http://www.pnas.org/content/99/12/7821
- A Fast Kernel-based Multilevel Algorithm for Graph Clustering, I. Dhillon, Y. Guan, and B, Kulis, 2005 http://www.acm.org/sigs/sigkdd/kdd2005/index.html

LIST OF SOURCES (CONT.)

- Modularity and community structure in networks (Newman, 2006)
- Community Detection Algorithms: A comparative analysis by Lancichinetti and Fortunato, 2009
- Community detection in graphs by Fortunato, 2009