Finding Communities by Clustering a Graph into Overlapping Subgraphs

Jeffrey Baumes, Mark Goldberg,
Mukkai Krishnamoorthy,
Malik Magdon-Ismail, Nathan Preston

Rensselaer Polytechnic Institute, Troy, NY, USA

Rensselaer

Outline

Introduction

Algorithms IS and RaRe

Experiments

Conclusions and future work

Rensselaer

Outline

Introduction

Algorithms IS and RaRe

Experiments

Conclusions and future work

Rensselaer

What is a cluster?

A *cluster* is a set of closely related objects, not as closely related to the rest

Clustering is the development of a collection of clusters

Traditionally, clustering is restricted to *partitioning* into non-overlapping clusters

Rensselaer

Overlapping clusters

Partitioning is well researched; many algorithms and software packages exist

(Kernighan-Lin, *k*-means, hierarchical, CHACO)

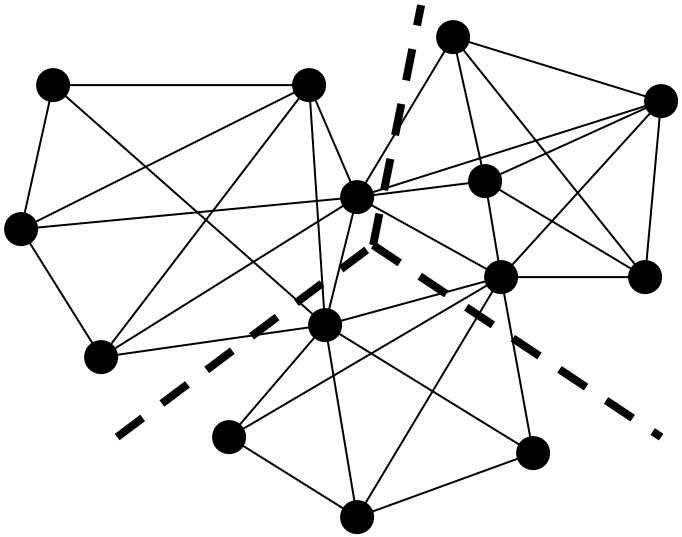
General clustering allows overlapping clusters

Such clustering is natural in social networks

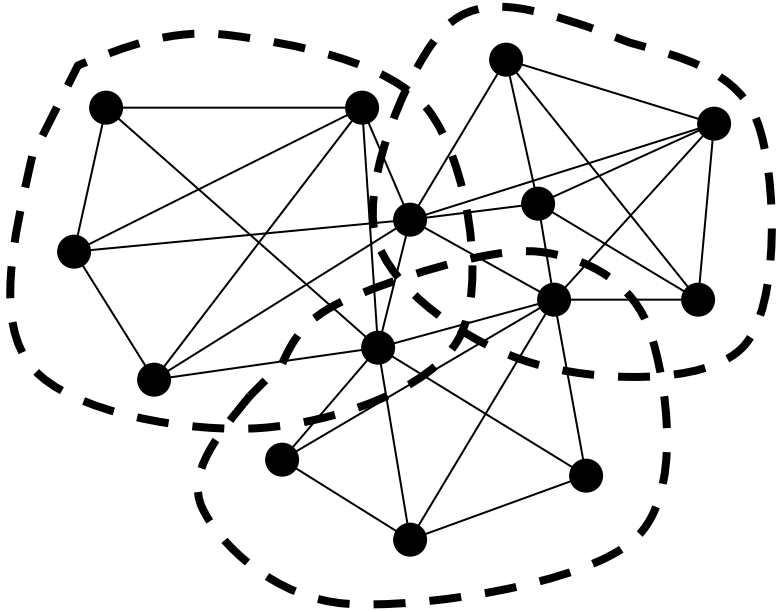
Clustering in a general sense is not well-studied

Example

Partitioning



General Clustering



A new definition of a cluster

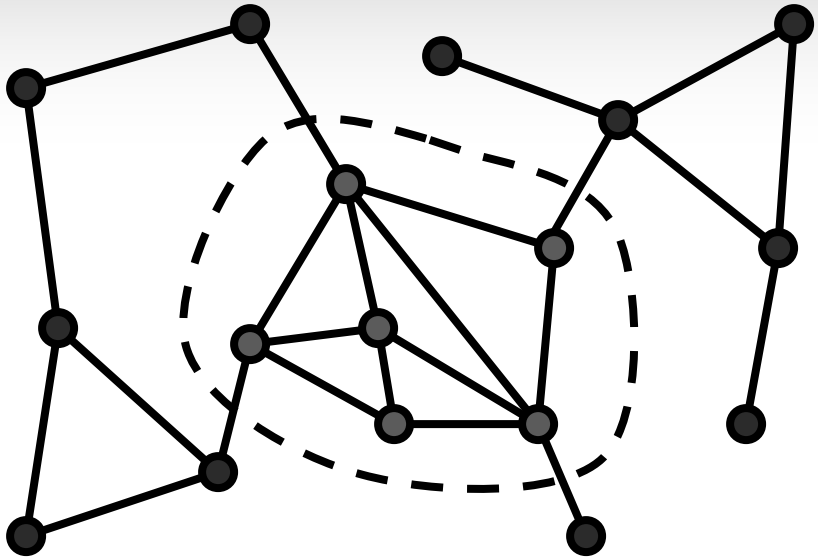
Define a *weight function* (or *density*) $W(C)$ for every subset of objects, then maximize W locally

A cluster is defined as a set of objects whose weight is larger than any set close to it; the cluster is said to be *locally optimal*

A set is "close" to another if it may be derived from the other by adding or removing one object

Rensselaer

Weighting functions



$$p_{ex} = \frac{\text{IIII}}{\text{⁞⁞⁞ × ⁞⁞⁞⁞⁞}}$$

$$\frac{\cancel{\text{IIII}}\ \cancel{\text{IIII}}}{\frac{1}{2}(\text{⁞⁞⁞ × ⁞⁞⁞})} \qquad \frac{\cancel{\text{IIII}}\ \cancel{\text{IIII}}}{\cancel{\text{IIII}}\ \cancel{\text{IIII}} + \text{IIII}} \qquad \frac{p_{in}}{p_{in} + p_{ex}} \qquad \frac{\Delta\Delta\Delta\Delta\Delta}{\frac{1}{6}(\text{⁞⁞⁞ × ⁞⁞⁞ × ⁞⁞})}$$

$W_p = p_{in}$ $\qquad$ W_e $\qquad$ W_i $\qquad$ W_Δ

Outline

Introduction

Algorithms IS and RaRe

Experiments

Conclusions and future work

Rensselaer

Iterative Scan (IS) algorithm

Begins with some cluster "seed"

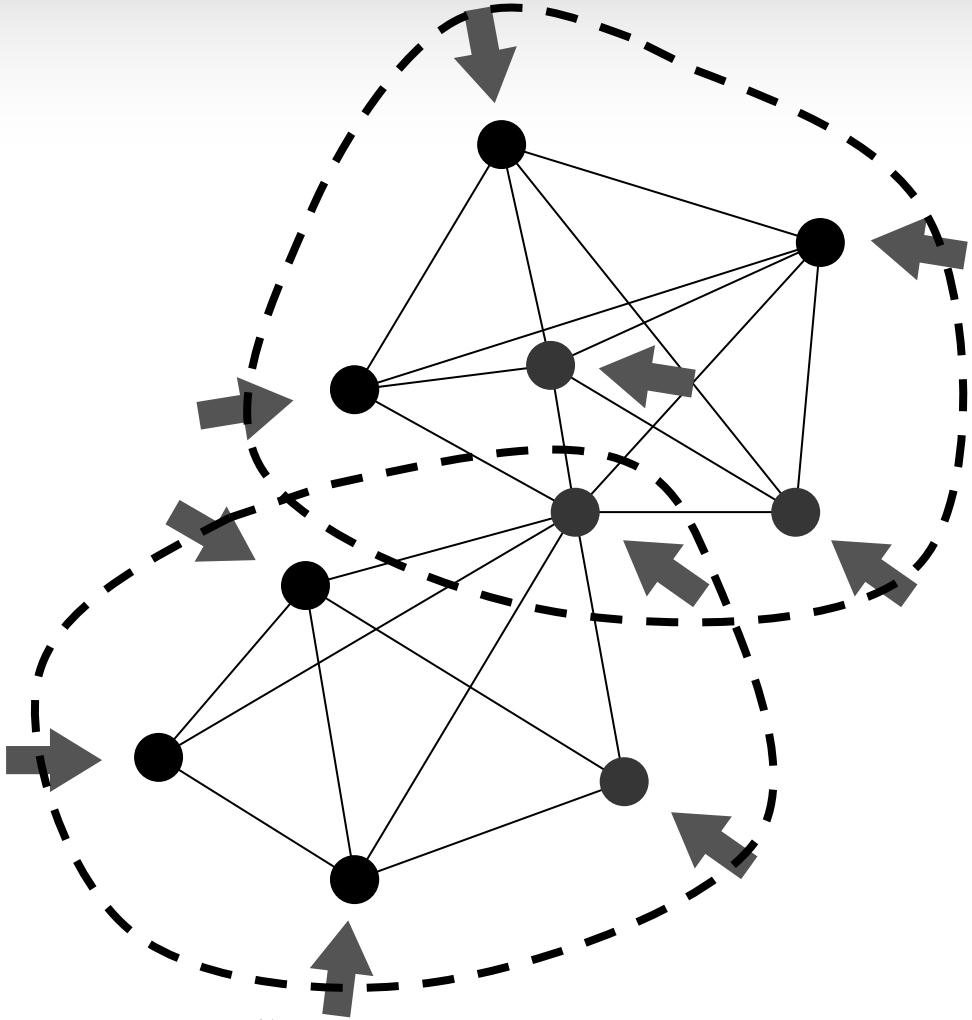
Traverses nodes, adding or removing the node while the cluster weight improves

Works for *any* choice of weight metric *W*

```
procedure IS(seed,G,W)
C ← seed; w ← W(C);
increased ← TRUE;
while increased do
  for all v ∈ V(G) do
    if v ∈ C then
      C′ ← C \ {v};
    else
      C′ ← C ∪ {v};
    if W(C′) > W(C) then
      C ← C′;
  if W(C) = w then
    increased ← FALSE;
  else
    w ← W(C);
return C;
```

Rensselaer

IS algorithm



Rensselaer

Seed clusters

IS depends on having good seed clusters

These clusters should represent the entire collection of objects

One option: use an existing partitioning algorithm to create the seed clusters

Another option: create a new algorithm with a global view to create seed clusters (possibly overlapping)

Rank Removal (RaRe) algorithm

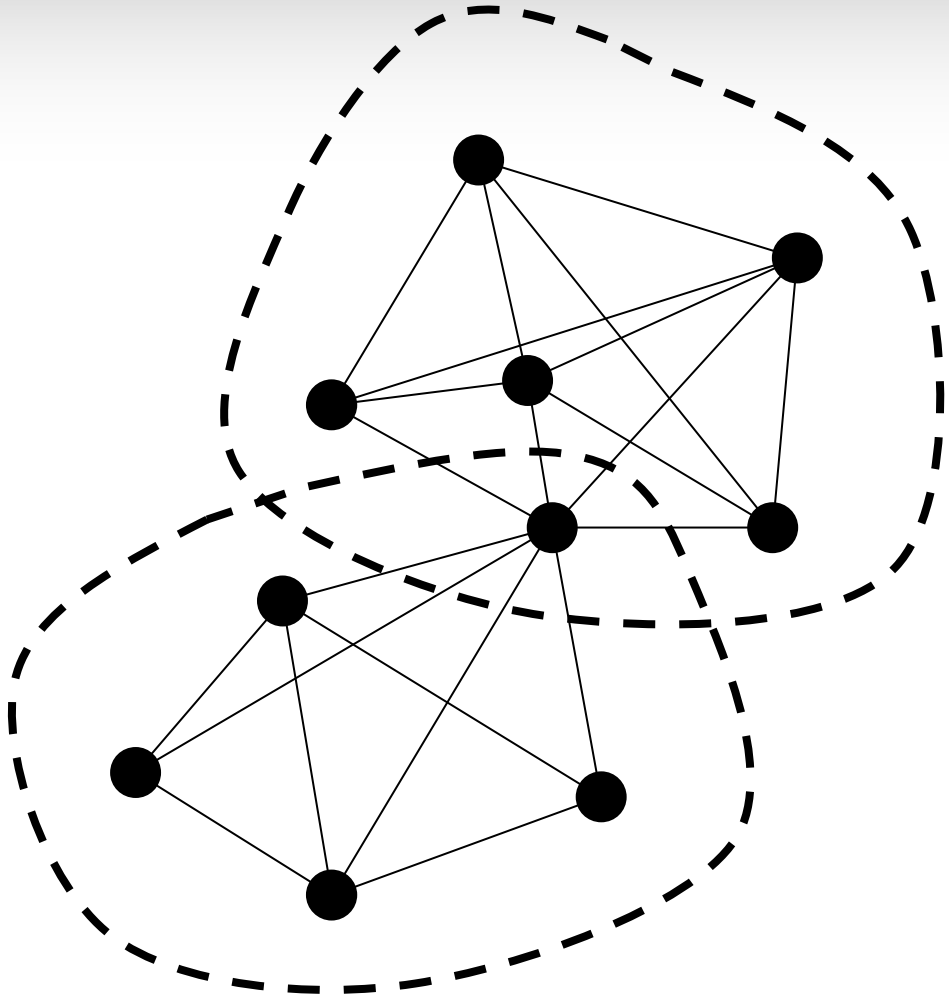
Idea: split the graph into components by removing a few key players (high-rank nodes)

Then, add the nodes back into whatever clusters they improve (possibly more than one)

procedure RaRe(C, W)
global $R \leftarrow \emptyset$;
$\{H_i\}$ are connected components in G;
for all H_i **do**
  ClusterComponent(H_i);
Initial clusters $\{C_i\}$ are cluster cores;
for all $v \in R$ **do**
  for all Clusters C_i **do**
    Add v to cluster C_i if v is adjacent to C_i or $W(v \cup C_i) > W(C_i)$;

procedure ClusterComponent(H)
if $|V(H)| > max$ **then**
  $\{v_i\}$ are t highest rank nodes in H;
  $R \leftarrow R \cup \{v_i\}$; $H \leftarrow H \setminus \{v_i\}$;
  $\{F_i\}$ are connected components in H;
  for all F_i **do**
    ClusterComponent(F_i);
else if $min \leq |V(H)| \leq max$ **then**
  mark H as a cluster core;

RaRe algorithm



Rensselaer

k-Neighborhood algorithm

A naïve overlapping clustering procedure

Selects random cluster centers

Cluster contains all nodes distance at most *k* from the center

"Baseline" for comparison with IS, RaRe

Rensselaer

Outline

Introduction

Algorithms IS and RaRe

Experiments

Conclusions and future work

Rensselaer

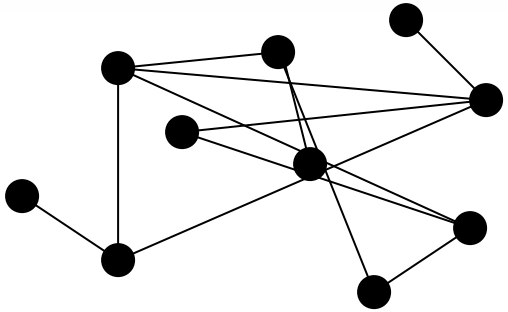
Experiments

Runtime and quality analysis

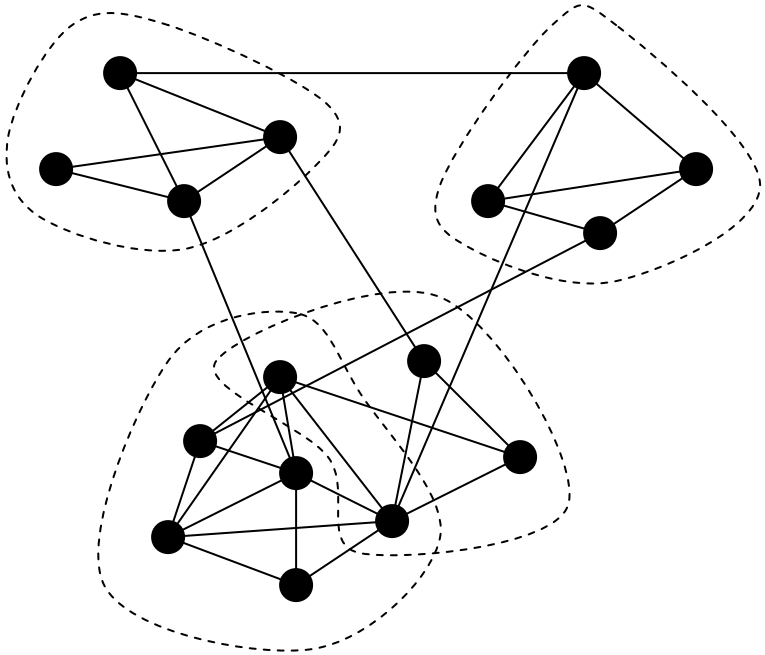
Compare algorithms on both real-world and simulated inputs

Simulated input

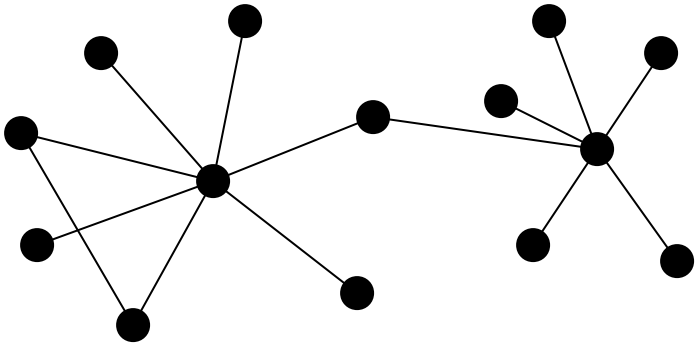
Random



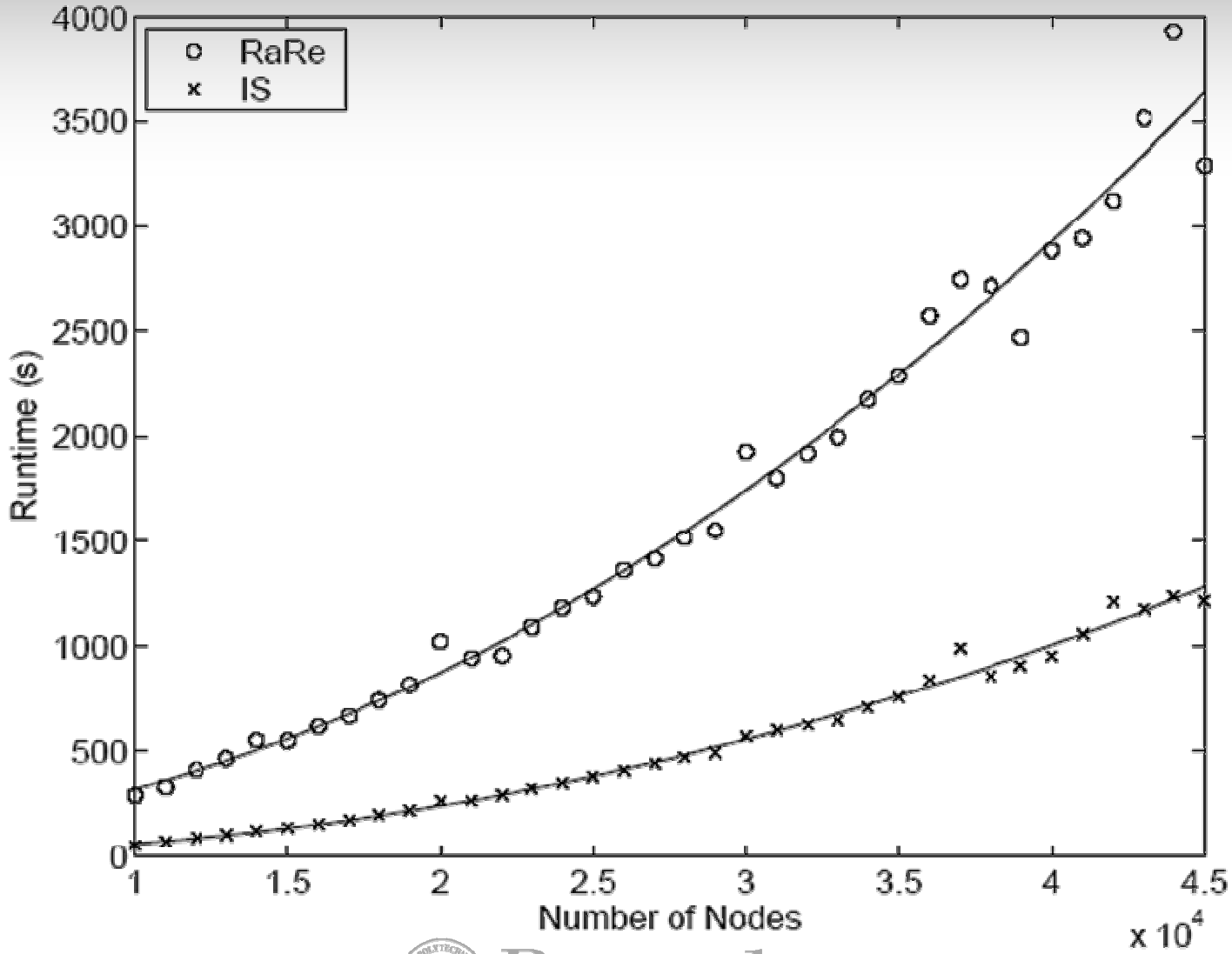
Group random



Preferential attachment

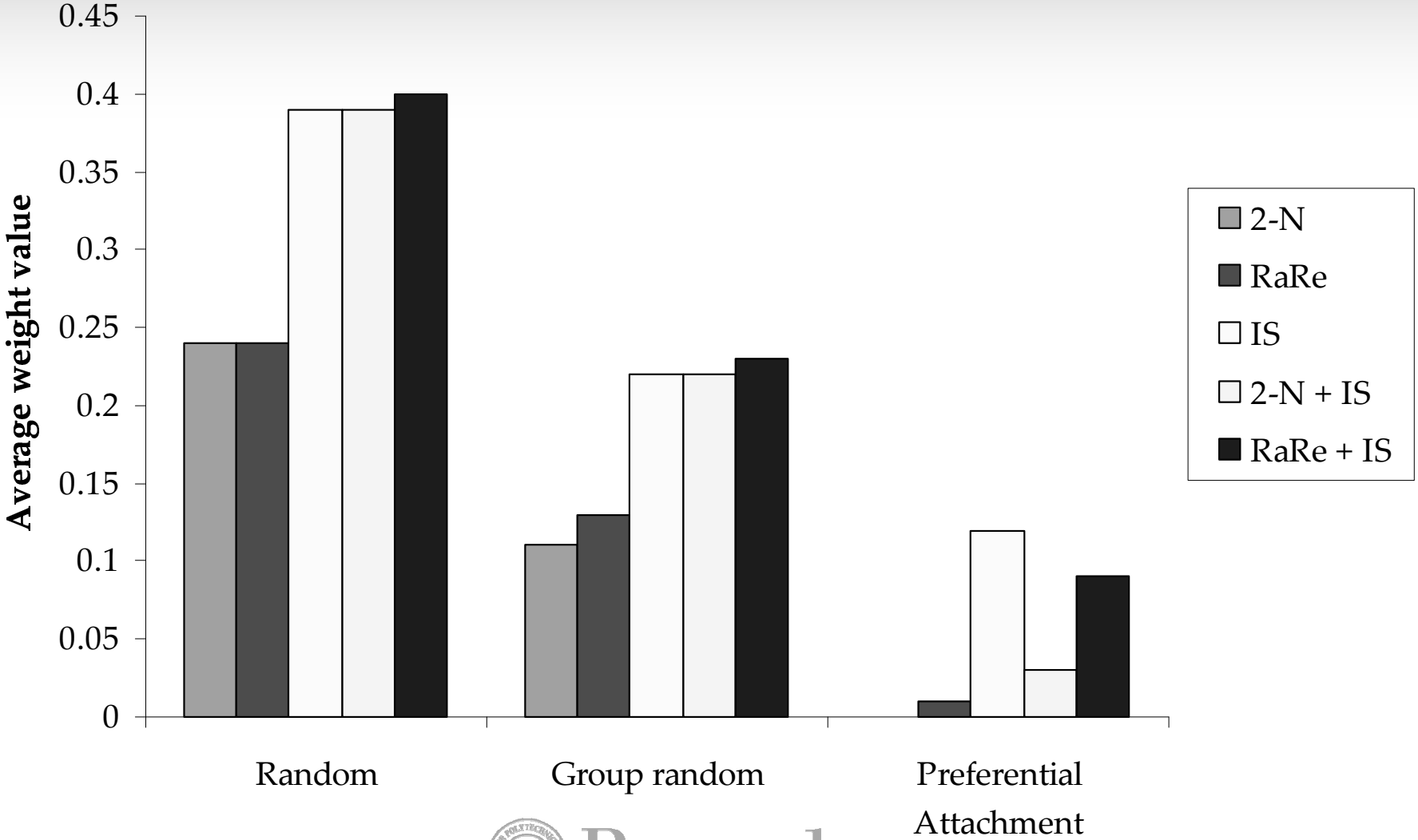


Runtime for simulated input



Rensselaer

Quality for simulated input



Rensselaer

Real-world graphs

575,600 node subset of the CiteSeer database

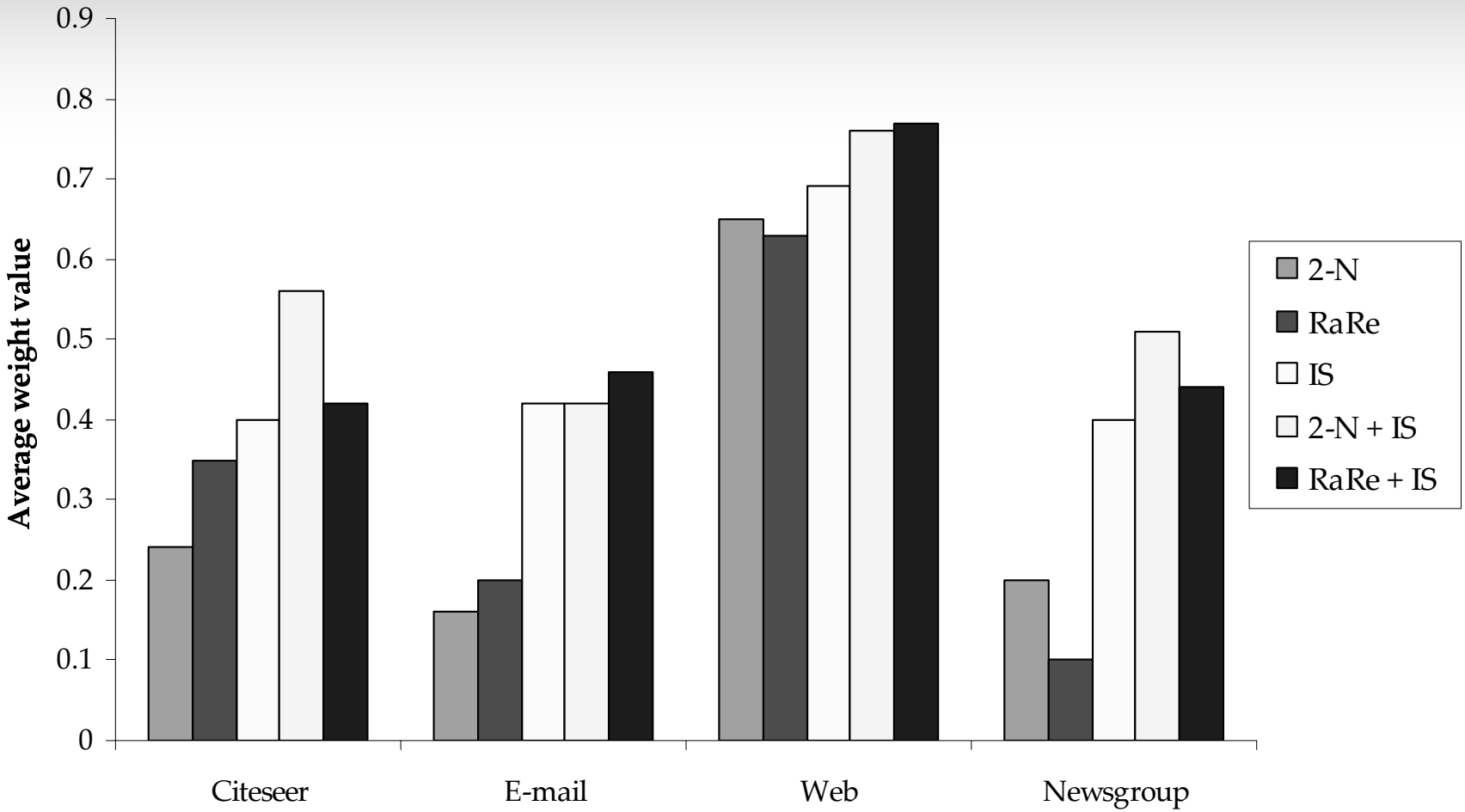
E-mail communications among RPI community over two days

Web graph of Malik Magdon-Ismail's website (www.cs.rpi.edu/~magdon)

Newsgroup posts on alt.conspiracy

Rensselaer

Quality for real-world graphs



Rensselaer

Outline

Introduction

Algorithms IS and RaRe

Experiments

Conclusions and future work

Rensselaer

Conclusions

First overlapping clustering algorithms successfully developed and tested

Local optimality is an intuitive criterion for overlapping clusters

RaRe/2-N improved by IS was best overall

Rensselaer

Future work

More robust testing with various weight metrics, and with known clusters

Develop new, more efficient algorithms for clustering

Detect communities that evolve over time

Rensselaer