Scaling Overlapping Clustering

Kyle Kloster
(now at NCSU)

Merrielle Spain
Stephen Kelley

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

This work is sponsored under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

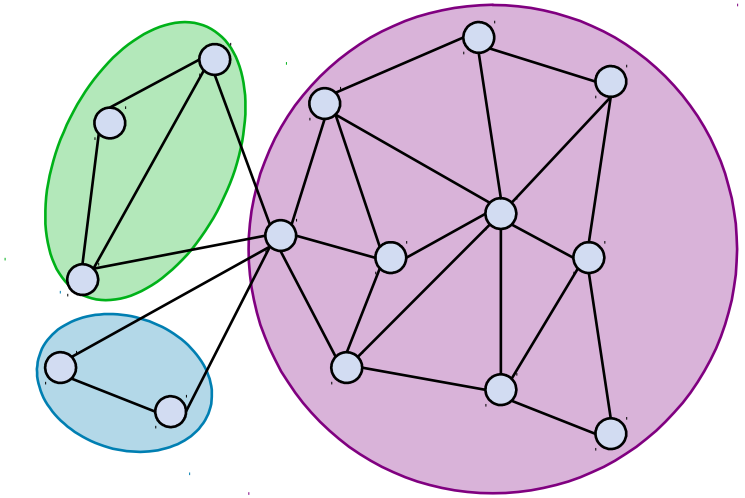
Scaling graph analytics

Broader Problem

- **Many global graph analytics scale poorly to large graphs**

Case study

- **Speed up overlapping clustering:**
  - **Link Clustering**
  - **Baselines: Infomap, BigCLAM**



[Ahn, Bagrow, Lehmann, Nature 2010] , [Rosvall, Bergstrom, PNAS 2008] , [Yang, Leskovec, ICDM 2013]

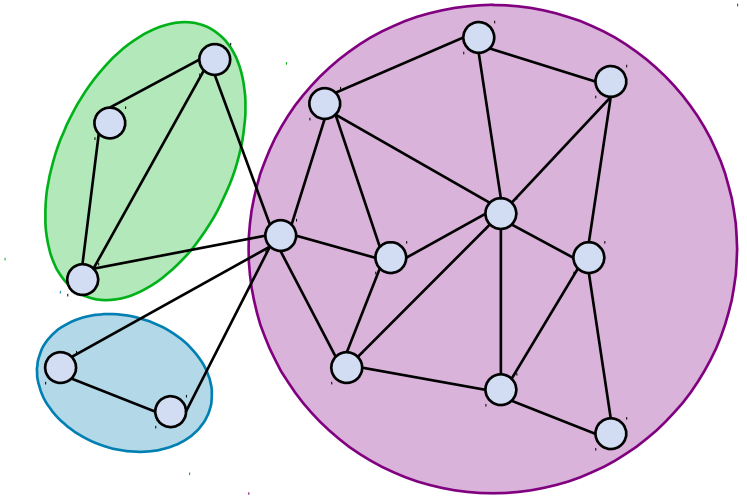
Scaling graph analytics

Broader Problem

- **Many global graph analytics scale poorly to large graphs**

Case study

- **Speed up overlapping clustering:**
  - **Link Clustering**
  - **Baselines: Infomap, BigCLAM**



Our problem

- **Even state-of-the-art algorithms scale poorly**

This talk

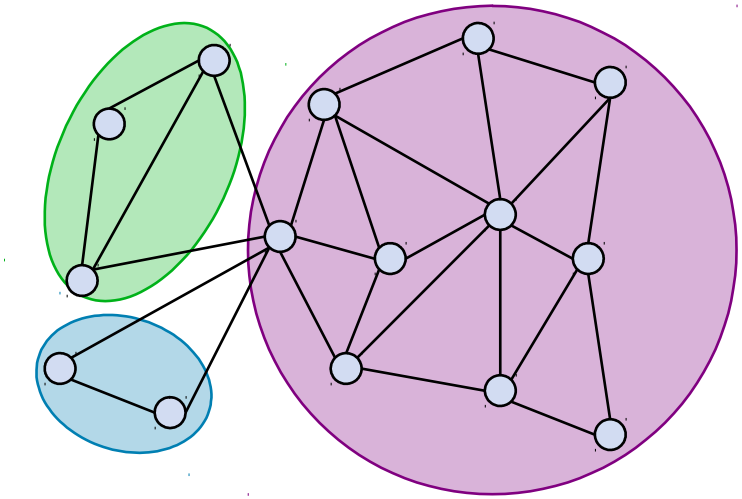
- **Tight complexity analysis -> identify, alleviate bottlenecks**

[Ahn, Bagrow, Lehmann, Nature 2010] , [Rosvall, Bergstrom, PNAS 2008] , [Yang, Leskovec, ICDM 2013]

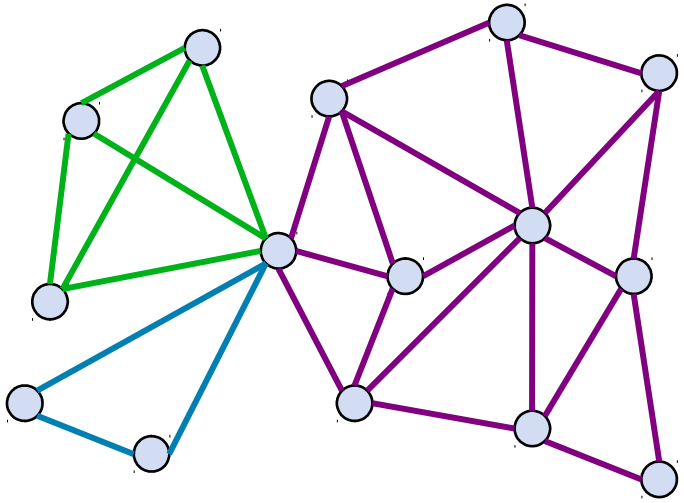
Our focus: Link Clustering

- **Detect communities by partitioning the *edges***
- **A node belongs to all communities that its edges do!**

Node clustering

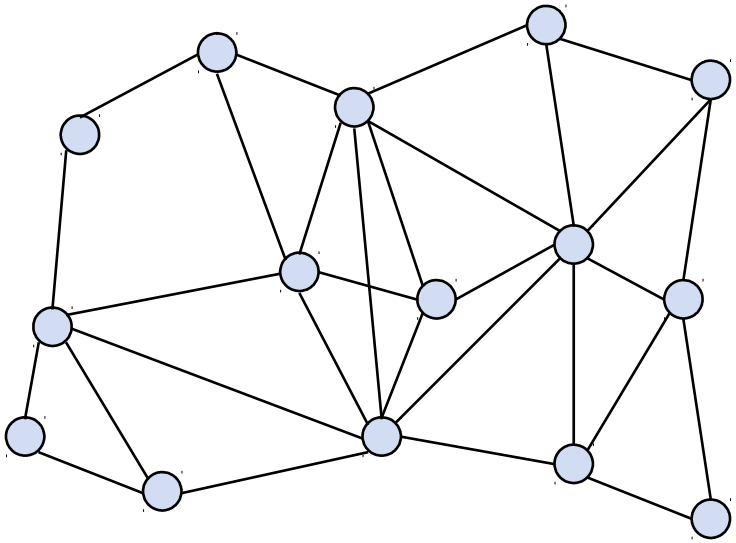


Link clustering



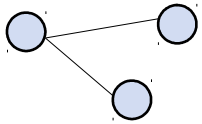
Ahn, Bagrow, Lehmann, "Link communities reveal multiscale complexity in networks." Nature, 2010.

Which property describes scaling?



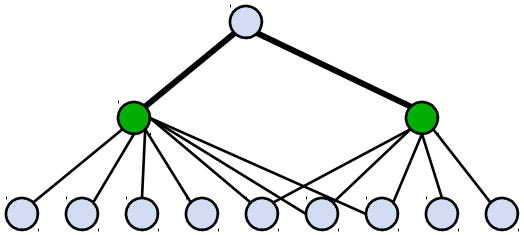
Features:

- **Edges**
- **Maximum node degree**
- **Nodes**
- **Wedges**



Link Clustering - Overview

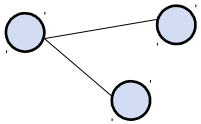
Calculate edge similarities → Sort similarities → Merge clusters → Partition



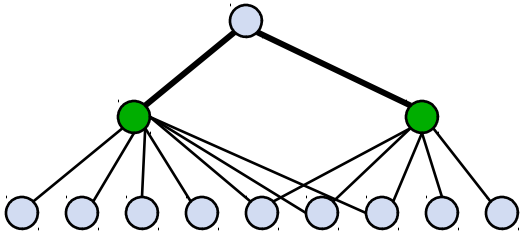
Ahn, Bagrow, Lehmann, “Link communities reveal multiscale complexity in networks.” Nature, 2010.

Link Clustering

Compute Jaccard Similarity of any 2 edges that share a node.



Calculate edge similarities → Sort similarities → Merge clusters → Partition



Jaccard Similarity:

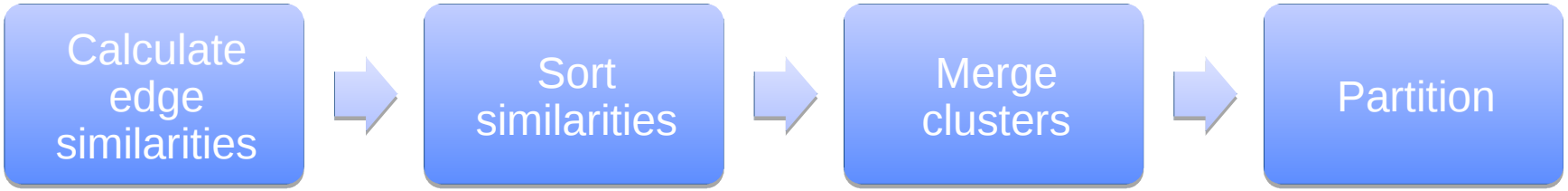
$$\frac{|\text{ neighborhood intersection }|}{|\text{ neighborhood union }|}$$

Ahn, Bagrow, Lehmann, "Link communities reveal multiscale complexity in networks." Nature, 2010.

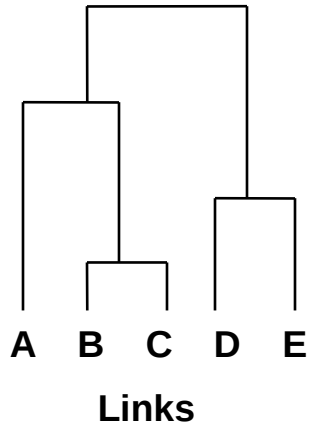
Link Clustering

Sort all pairs of edges (wedges) by their Jaccard Similarity.

Combine clusters with highly similar edges.

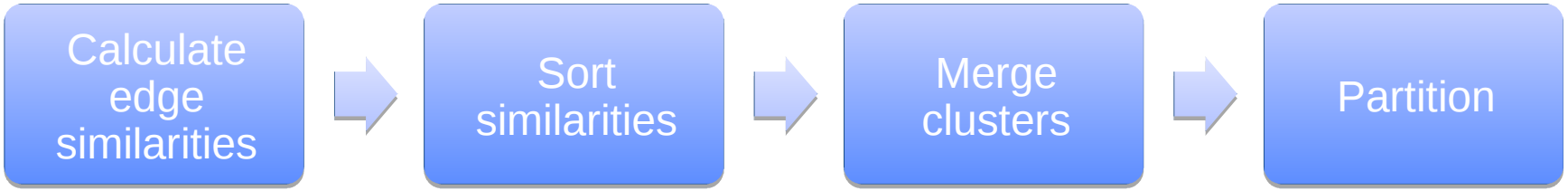


B,C
D,E
A,C
A,B
B,D
C,E
A,D
A,E
B,E
C,D



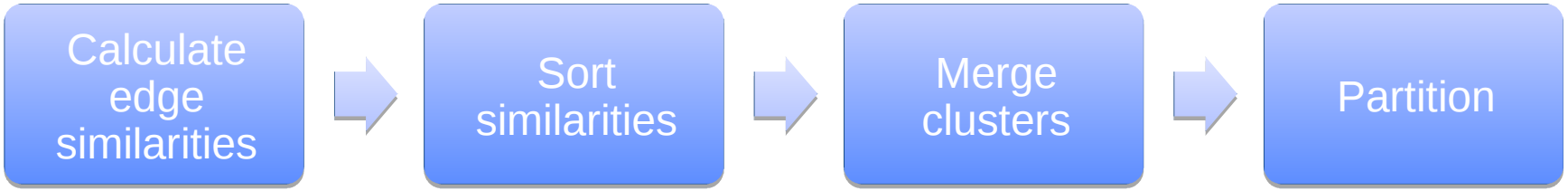
Ahn, Bagrow, Lehmann, "Link communities reveal multiscale complexity in networks." Nature, 2010.

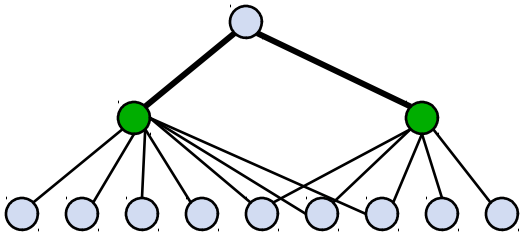
Link Clustering: tight complexity



$\mathcal{O}(wedges)$ $\mathcal{O}(wedges \times \log_2(wedges))$

Link Clustering





Number of wedges at a node =

(degree choose 2) ~ degree^2

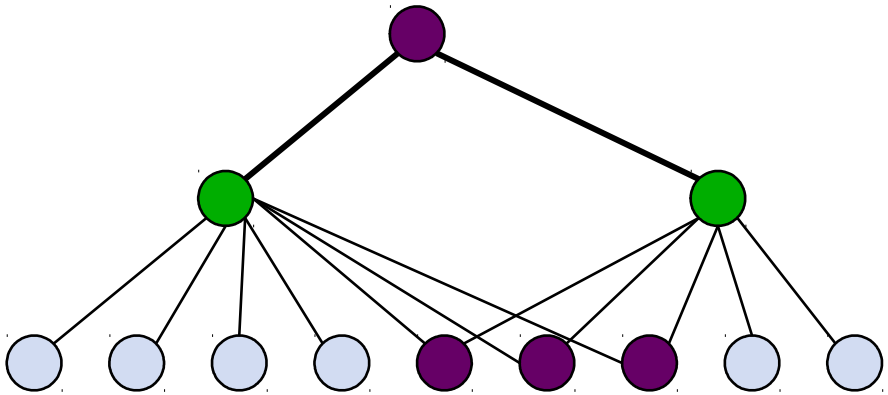
Large degree nodes (hubs) bad...

Ahn, Bagrow, Lehmann, "Link communities reveal multiscale complexity in networks." Nature, 2010.

Tight complexity identifies bottlenecks

- ***Sorting*** **= dominant subroutine**
- **Runtime scales with wedges …**
  - **Hubs cause ~O(n^2) wedges!**

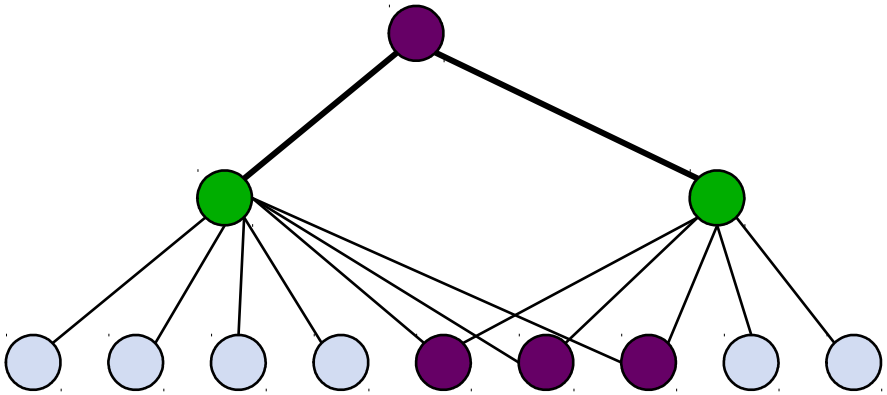
Bottlenecks: hubs, sorting



Degree is the maximum degree (number of neighbors) in graph

Similarity scores are structured

Jaccard Similarity:



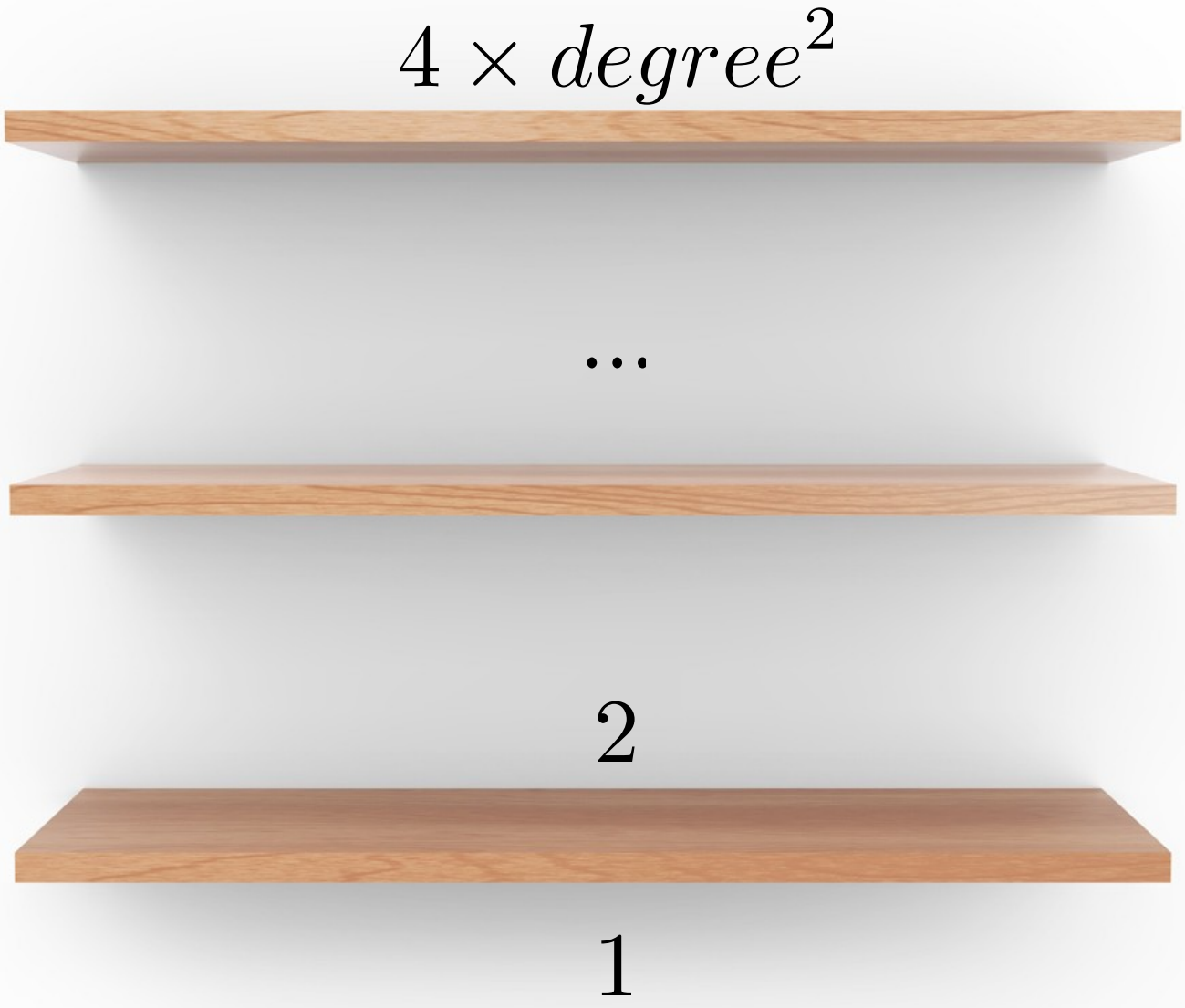
- **is a fraction: denominator, numerator are integers in**

  [1 , 2*max degree]

- **We show this implies small number of distinct scores...**
- **Enables linear-time sort!**
- **“Shelf” sort: bucket/counting**

$$\mathcal{O}(wedges \times \log_2(wedges)) \longrightarrow \mathcal{O}(wedges)$$

Shelf Sort



K. Kloster, D.F. Gleich (2015) Personalized PageRank Solution Paths. http://arxiv.org/abs/1503.00322

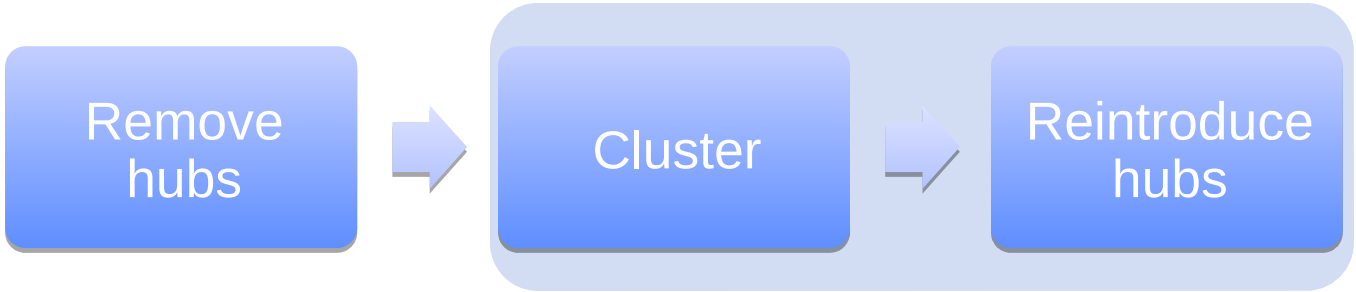
Handling Hub Bottlenecks

- **Claim: large degree nodes (hubs) can be removed**
  - **often meaningless (Spammers)**
  - **often uninformative (Twitter: Bieber, Obama; bio examples)**
  - **if important, worth processing individually**

- **Hubs drive up runtime**
  - **Hub -> ~n^2 wedges**

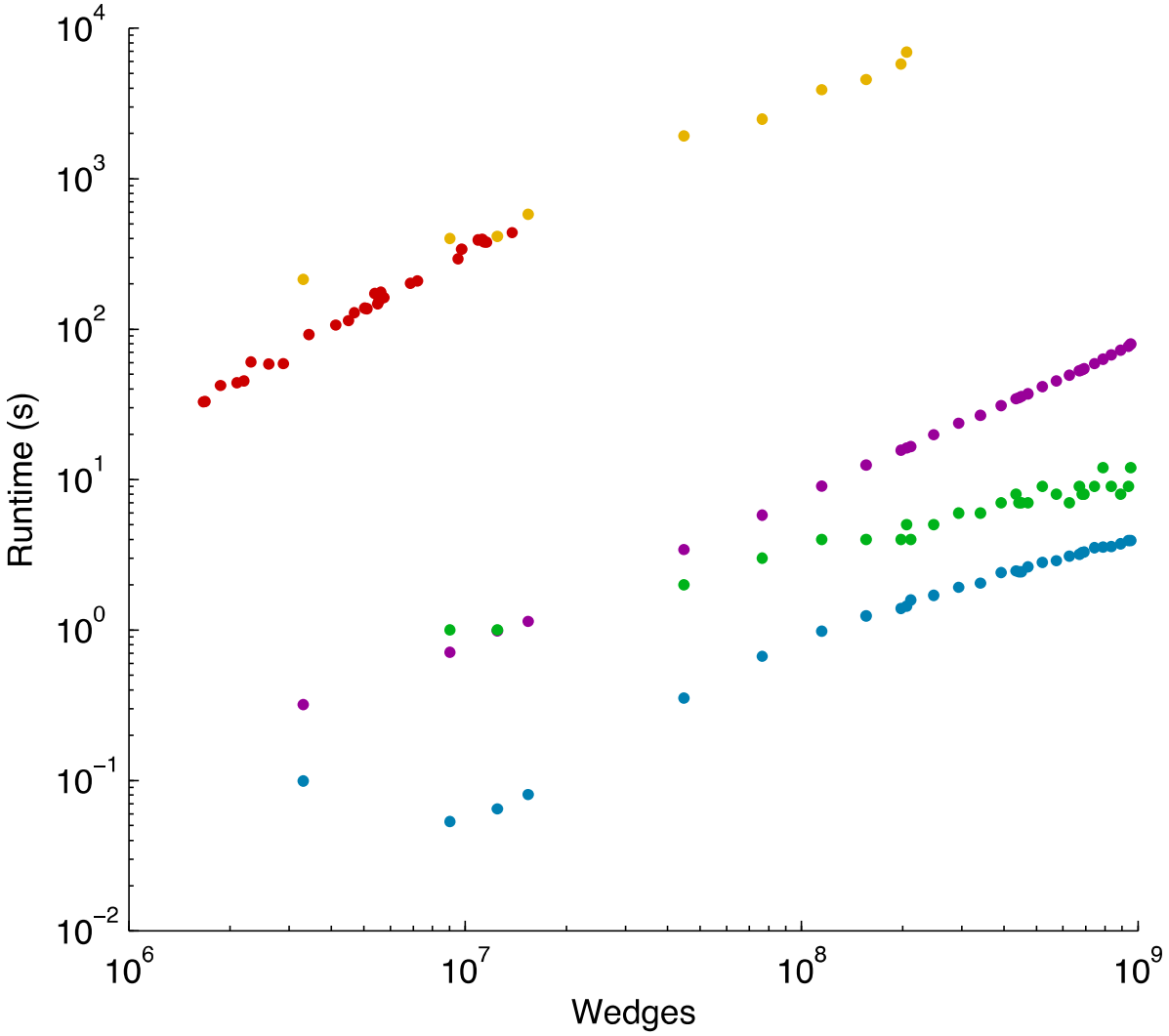
- **Hubs can obscure smaller-scale community structure**

Hub Removal



- **Add hubs only to clusters they strongly connect to**

~10,000 X faster



Preserves quality of clustering!

Conclusions

Scaling graph analytics:

- **Tighter complexity analysis can reveal bottlenecks**
  - **Runtime = O(sorting wedges)**
  - **Dominant subroutines, problematic graph features**
- **Exploiting bottleneck analysis**
  - **Remove bottleneck graph feature; handle separately?**
  - **Graphs are discrete -> structure! (e.g. sorting = linear)**
- **Context can guide handling bottleneck**
  - **are your bottlenecks disposable? (spam, Justin Bieber)**
  - **… worth processing separately?**