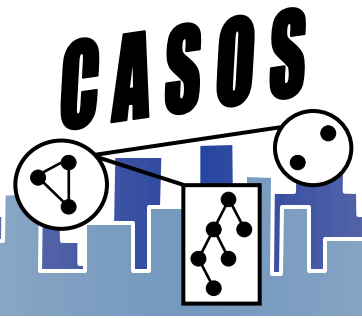


Large Dynamic Networks: Lessons and Thoughts

Geoffrey P. Morgan
Matthew Benigni

institute for SOFTWARE RESEARCH

Carnegie Mellon

Center for Computational Analysis of
Social and Organizational Systems
http://www.casos.cs.cmu.edu/

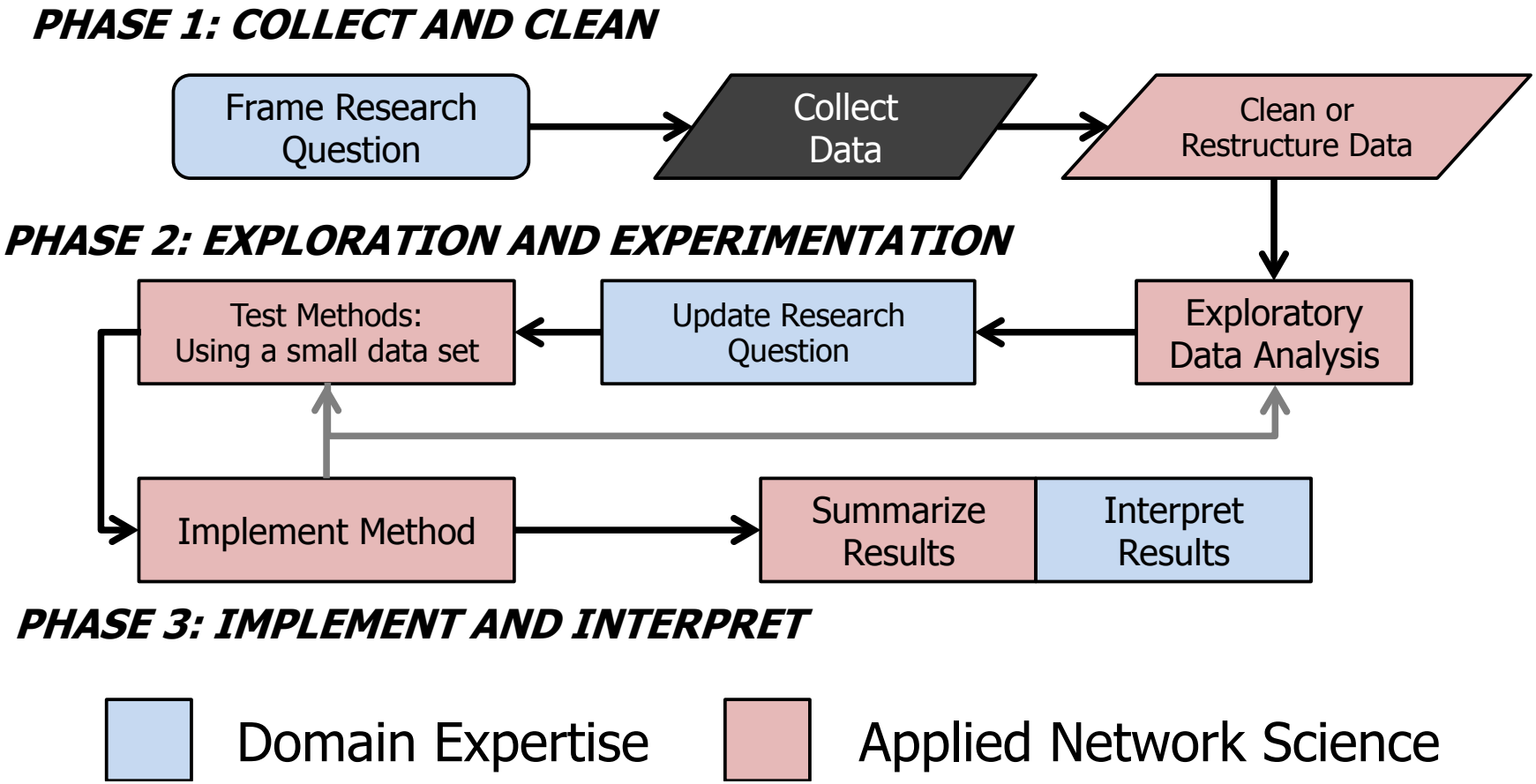
Introduction

- Large Network Analysis, as a very specific and very powerful form of "Big Data" analysis, is compelling and interesting, but it has its challenges.

- When we say "Large Network", we mean
  - where there are significantly more than 10,000 nodes
  - Frequently, 1M+ nodes are common

- Networks this large introduce unique constraints on your analysis.

The Big Data Pipeline



Sources of Data

- Frequently, Social Media Data is leveraged to get large-datasets.
- However, many large governmental entities produce event data sheets that can be conceptualized as a network, and fall into this context.

- Often, large network data is event data.
  - A retweeted B
  - C filed a case against D
  - E emailed F

Data Structure

$$R = \begin{bmatrix} \mathbf{r_{1,1}} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & \mathbf{r_{2,2}} & & \vdots \\ \vdots & & \ddots & \vdots \\ r_{n,1} & \cdots & \cdots & \mathbf{r_{n,n}} \end{bmatrix}$$

vs.

Source, Target
Jeff, Geoff1, 12
Geoff2, Geoff5, 3
Geoff5, Jeff, 45
….

- Large networks are typically sparse (i.e. not dense), and sparse matrices are more efficiently stored and manipulated with edge lists

Links drive the size of your network in terms of RAM and cost of matrix algebra, nodes drive the cost of many of the metrics and clustering algorithms

Why do these analyses take time? A Digression into Math

- There will be more on this later, but essentially, many traditional SNA measures offer insights but are also "expensive".

- In Computer Science, we express cost of an algorithm using "Big-O Notation", if you have *n* items to process
  - An incredibly slow process will require *n^n* operations
  - A very slow process will require *n!* operations
  - A slow process will require *n^2* operations
  - A fast process will require *n* operations
  - A very fast process will require *log n* operations

- Scientists who develop new algorithms are often trying to figure out clever ways of lowering the cost of getting good answers
  - Either exploiting the structure of the data
  - Or finding shortcuts that still allow for "good enough" answers

Algorithmic Complexity Matters

Algorithm	n = 4	n = 10	n = 100	N = 1,000
n^n	256	1 x E10	1 x E200	GINORMOUS!
n!	24	3.6M	9.3 x E157	RIDICULOUS!
n^2	16	100	10,000	1M
n	4	10	100	1,000
log n	1.39	2.30	4.61	6.91

Many traditional SNA measures (e.g., betweenness) were n^2.

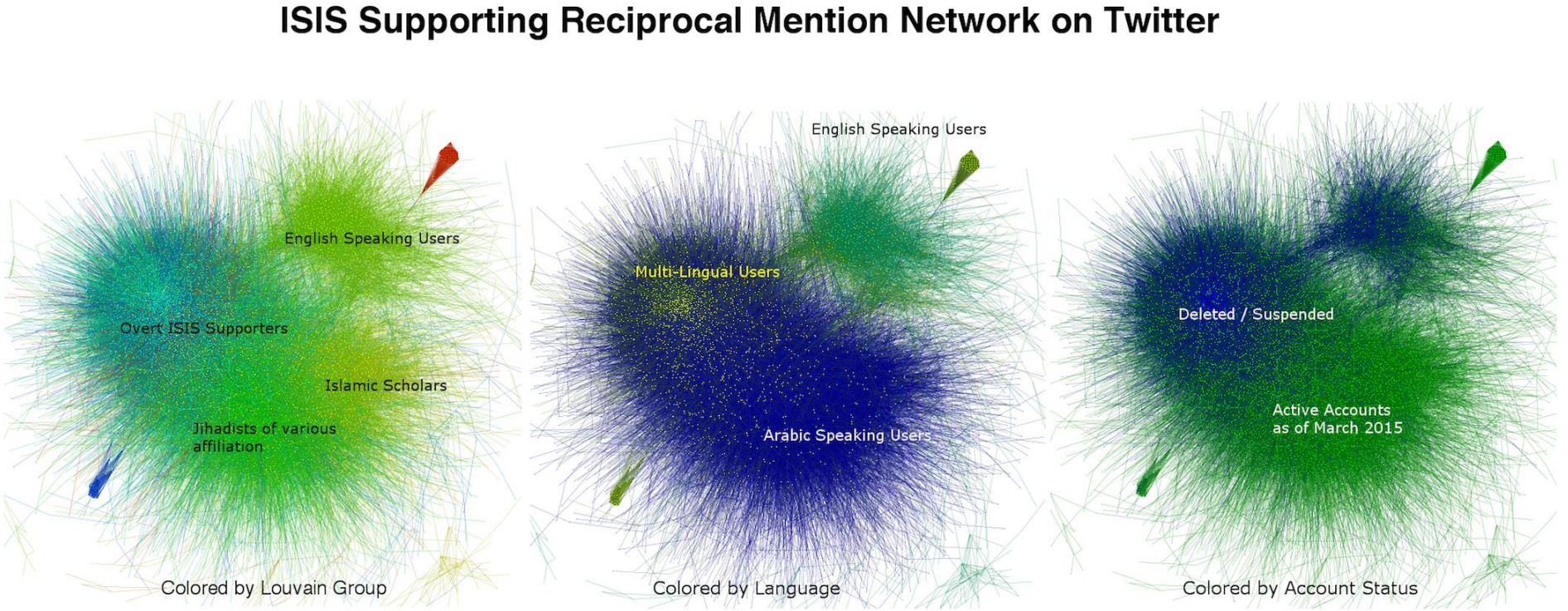
But it's not just Measure Calculation…

- Do you have ideas on other areas where a truly enormous network could cause data analysis problems?

Our Experiences

- Graphical Visualization
  - Standard lay-out algorithms are naively n^2.
  - Large networks tends to be "hair-balls"
- Node representation in the UI
  - Ability to interact with specific nodes requires memory
  - Every checkbox is itself a complicated object
- Manipulation of large files can be difficult
  - 32-Bit Excel doesn't support more than 1M rows
  - Many text editors refuse to open files larger than 4GB

Visualization



My immediate "solutions"

- Choose SNA measures wisely (see next talk)
- Don't visualize
  - But wait, see later thoughts
  - Also, CASOS is working on large data visualization
- Use ORA in Batch Mode
- Use scripts to work with my enormous data programmatically

But these solutions, especially #3, require programming expertise

Longer-Range Solutions

- Consider your representational form

Example: I have 2M+ Emails, sent between 12000 people, should I?

- Represent all 2M Emails as nodes, with people connected to emails?
- Connect people directly as link weights based on email activity?

Example 2: I have 300,000 people that are part of 12,000 communities, should I?

- Represent people and communities as nodes? With people connected to communities?
- Represent communities, with link weights between communities the number of people connected to both?

Longer Range Solutions, Continued

- Limit your analysis to what is actually of interest
  - Dynamics can be incredibly useful here
    - General Rule: The more you can slice up your data, the easier it is to work with (The Blender Maxim)
  - With very large networks, choose metrics wisely, K-Betweenness, for example, is probably better than Betweenness
  - Example: I am interested in the social role of a particular organization's VP based on his email activity, should I?
    - Evaluate everyone in the organization, and place the VP in ranked measure lists?
    - Use 2 or 3-distance neighborhood of the VP to characterize his local position?

Longer Range Solutions, Continued

- Leverage your data!
  - If you're using event data, that means you can pre-filter the events you pull from a given chronological period (Databases are pretty great for this)
  - Consider "constructing networks" as necessary based on a larger pools of data