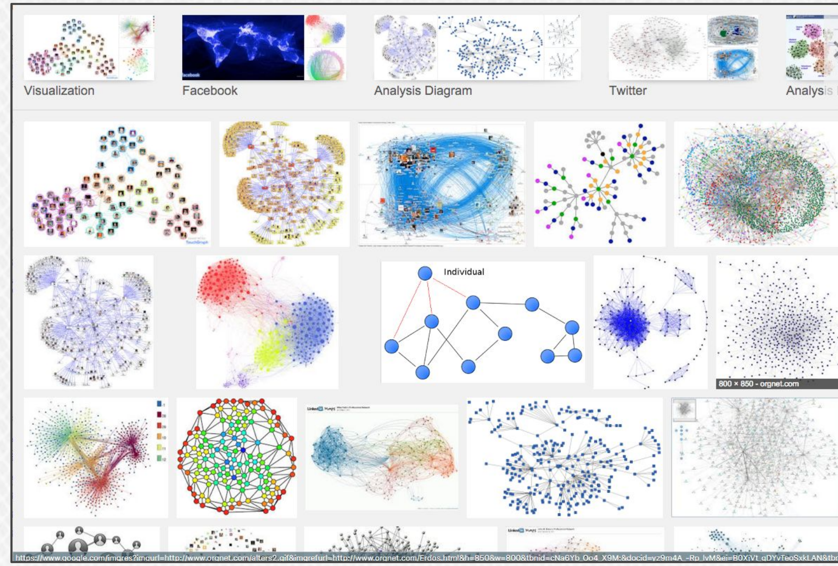
Graph Analysis with Python and NetworkX

DistrictDataLabs

Graphs and Networks

What is a graph?



graph[1]

/graf/

noun

1. a diagram showing the relation between variable quantities, typically of two variables, each measured along one of a pair of axes at right angles.
   synonyms: chart, diagram; More

verb

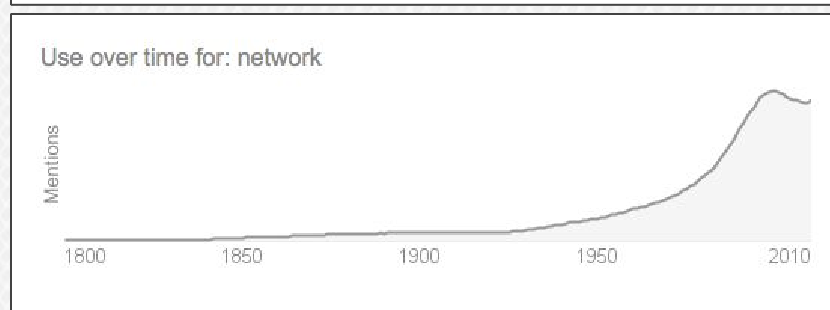
1. plot or trace on a graph.
   synonyms: plot, trace, draw up, delineate
   "we graphed the new prices"

net·work

/ˈnetˌwərk/

noun

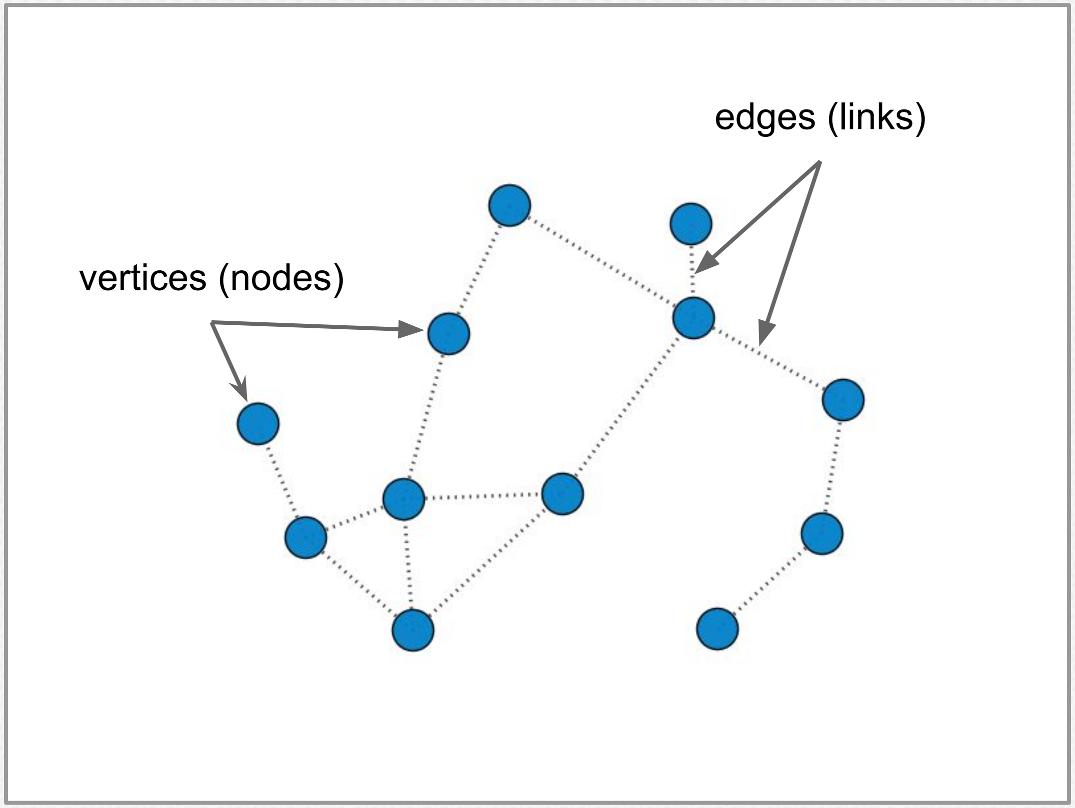
1. an arrangement of intersecting horizontal and vertical lines.
   synonyms: web, lattice, net, matrix, mesh, crisscross, grid, reticulum, reticulation; plexus
   "a network of arteries"
2. a group or system of interconnected people or things.
   "a trade network"
   synonyms: system, complex, nexus, web, webwork
   "a network of friends"



Graph Theory

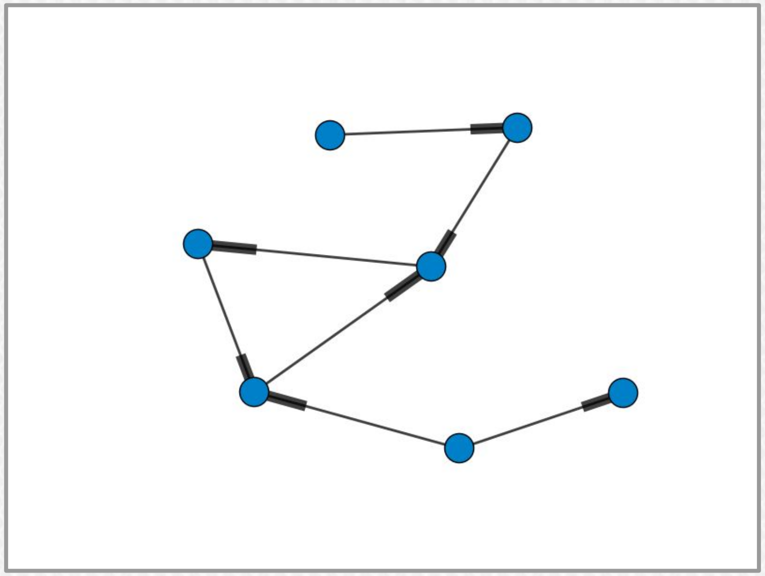
The Mathematical study of the application and properties of graphs, originally motivated by the study of games of chance.

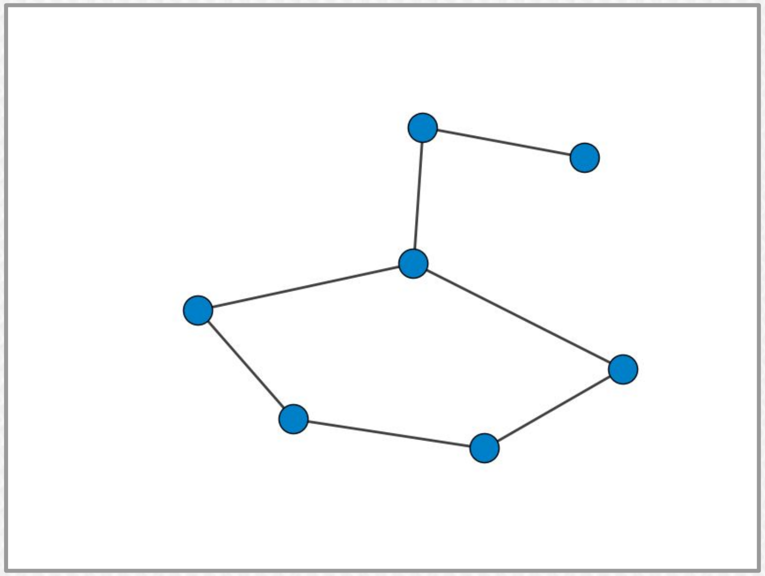
Traced back to Euler’s work on the Konigsberg Bridges problem (1735), leading to the concept of Eulerian graphs.



$$G=(V,E)$$

A Graph, `G`, consists of a finite set denoted by `V` or `V(G)` and a collection `E` or `E(G)` of ordered or unordered pairs `{u,v}` where `u` and `v` $\in$ `V`





Graphs can be directed or undirected

DiGraphs, the edges are ordered pairs: `(u,v)`

Network Definitions

$$G=(V,E) \quad E \subset V^2$$

Graphs as sets

$$\{(x,x) \mid x \in V\} \bigcap E = \varnothing$$

Local Cyclicity

Cardinality

$O(G)=|V|$ Order

$S(G)=|E|$ Size

Describing Graphs

Adjacency Matrix

$$G=(V,E) \quad E \subset V^2$$

$$A_{ij}=\begin{cases} 1 \ \ if \ \ (i,j) \in E \\ 0 \ \ otherwise \end{cases}$$

```
[[0, 0, 1, 1, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
 [1, 0, 0, 1, 1, 0, 0, 0, 0, 1],
 [1, 0, 1, 0, 0, 0, 1, 1, 1, 0],
 [0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 1, 0, 1, 0, 0, 0, 0, 1, 0],
 [0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 1, 0, 0, 1, 0, 0, 1],
 [0, 0, 1, 0, 0, 0, 0, 0, 1, 0]]
```

Representing Graphs

Adjacency Matrix

$$G=(V,E) \quad E \subset V^2$$

$$A_{ij}=\begin{cases} 1 \;\; if \;\; (i,j) \in E \\ 0 \;\; otherwise \end{cases}$$

```
[[0, 0, 1, 1, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
 [1, 0, 0, 1, 1, 0, 0, 0, 0, 1],
 [1, 0, 1, 0, 0, 0, 1, 1, 1, 0],
 [0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 1, 0, 1, 0, 0, 0, 0, 1, 0],
 [0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 1, 0, 0, 1, 0, 0, 1],
 [0, 0, 1, 0, 0, 0, 0, 0, 1, 0]]
```

Undirected graphs have symmetric adjacency matrices.

Representing Graphs

Node Neighbors

$$N_G(v_i) = (v_j \in A_{ij} \quad if \quad A_{ij} = 1)$$

Degree

$$K(v_i) = |N_G(v_i)|$$

Directed Networks

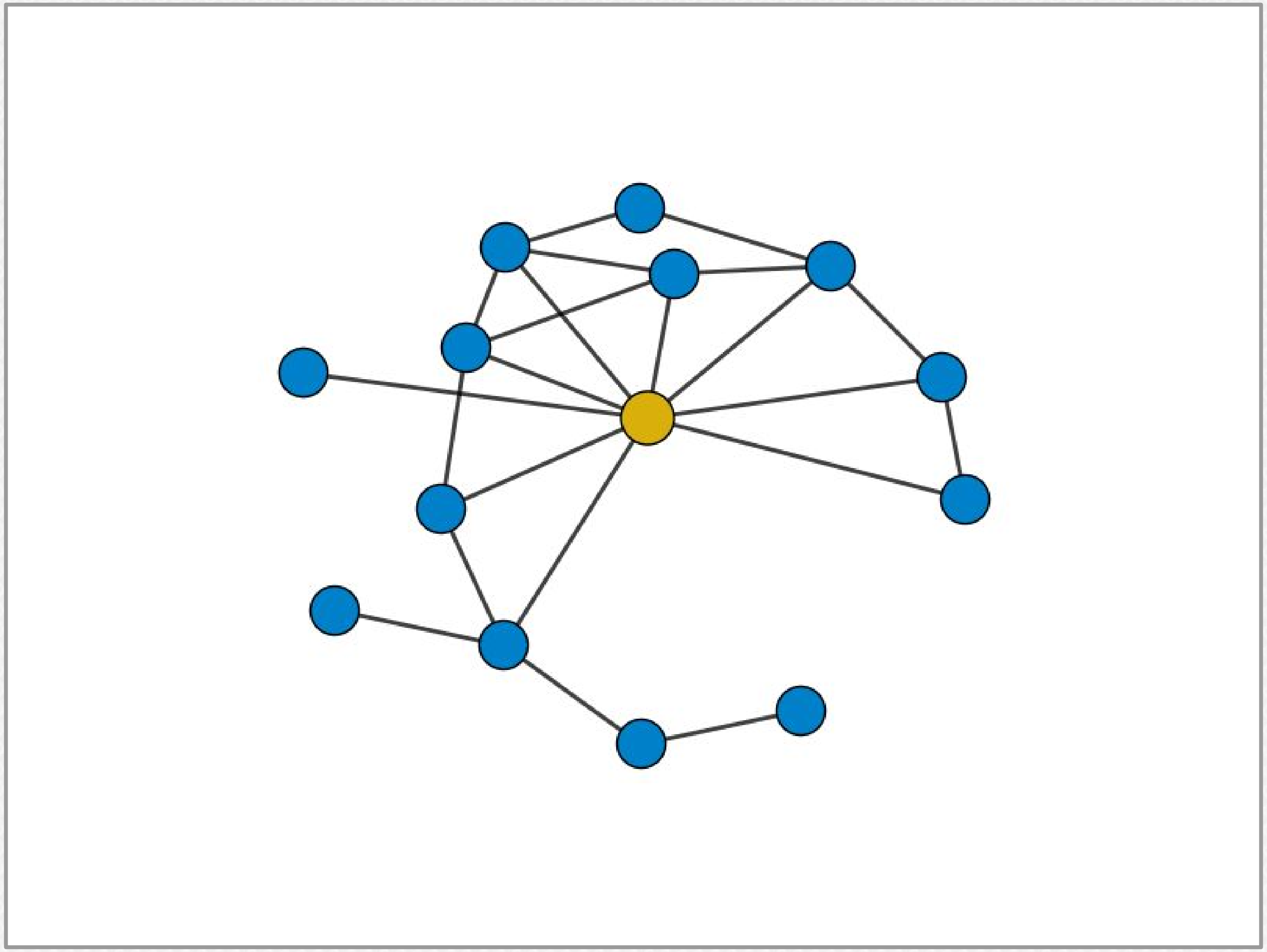
$$k_i^{out} = \sum_j A_{ij} \qquad k_i^{in} = \sum_j A_{ji}$$

$$k_i = k_i^{in} + k_i^{out}$$

Undirected Networks

$$k_i = \sum_j A_{ji} = \sum_j A_{ij}$$

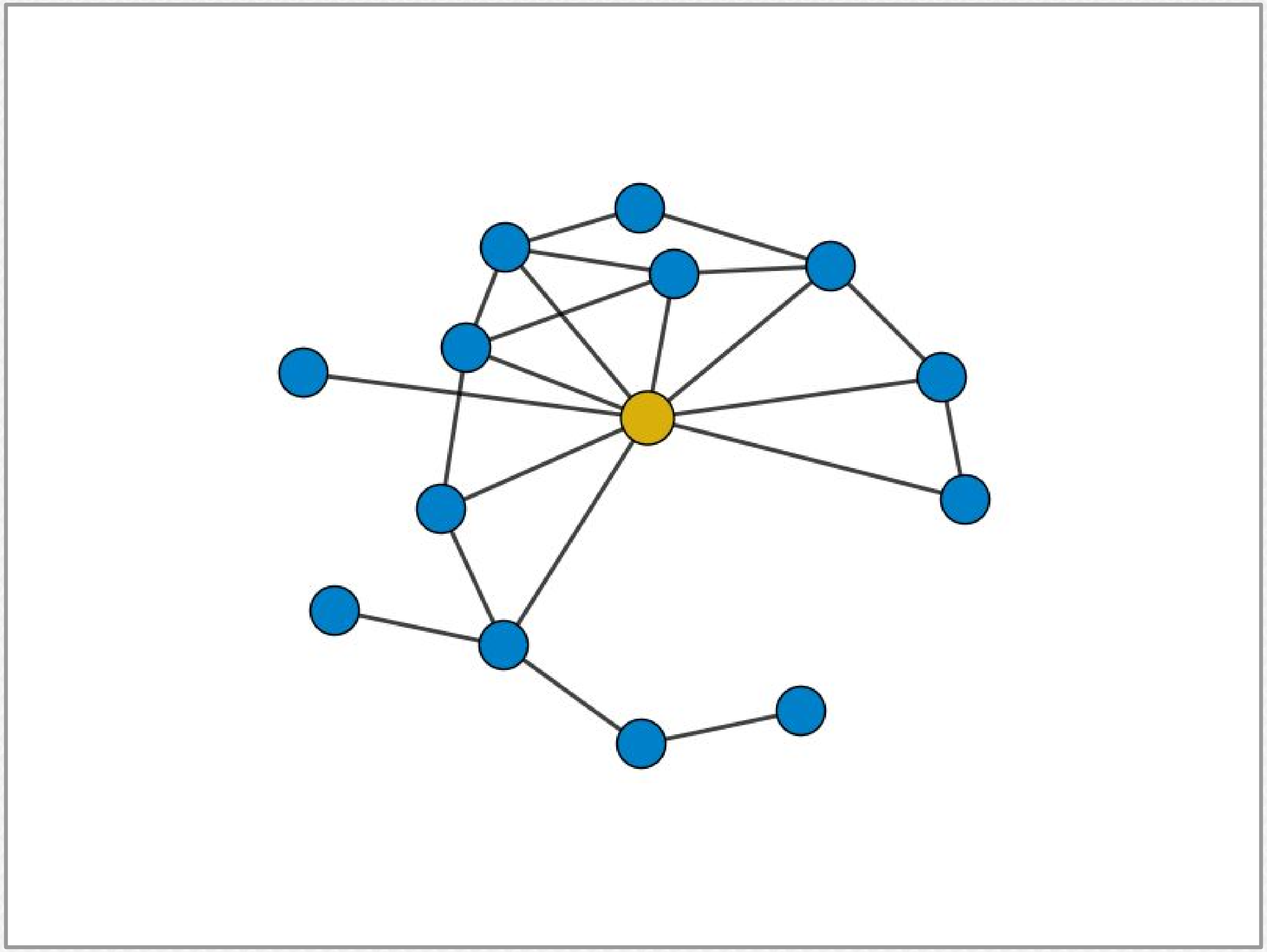
Describing Vertices



$O(G) = ?$

$S(G) = ?$

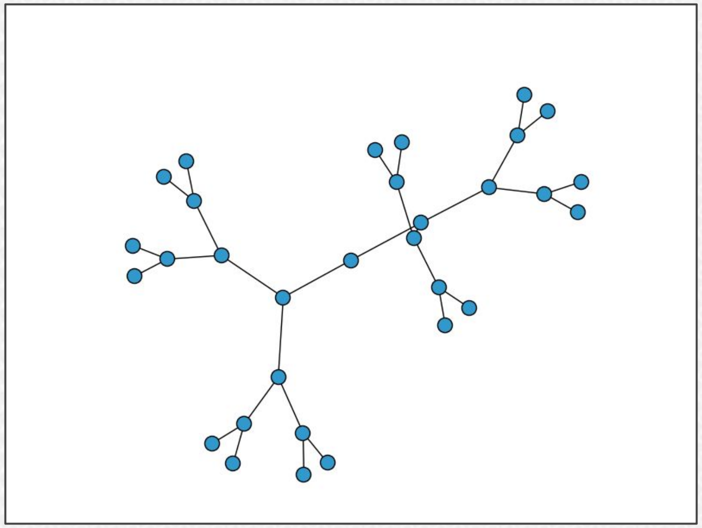
$k_Y = ?$



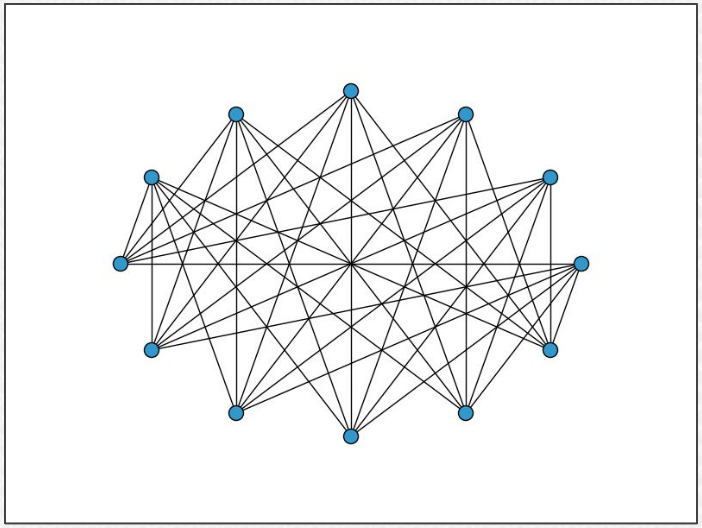
$O(G) = 14$

$S(G) = 21$

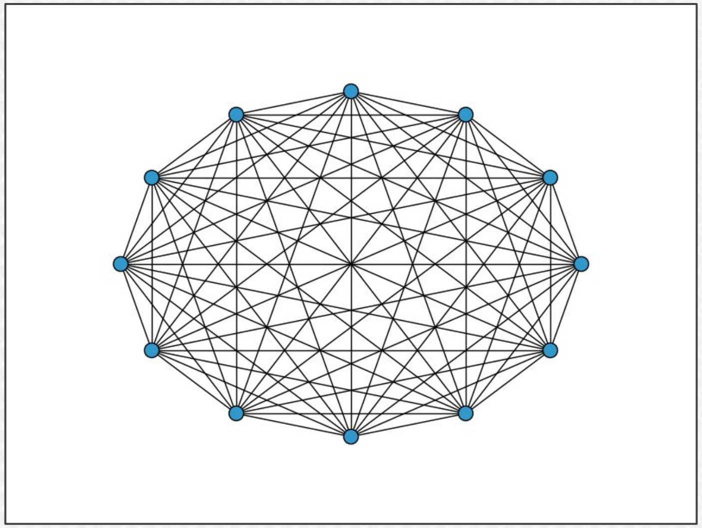
$k_Y = 9$



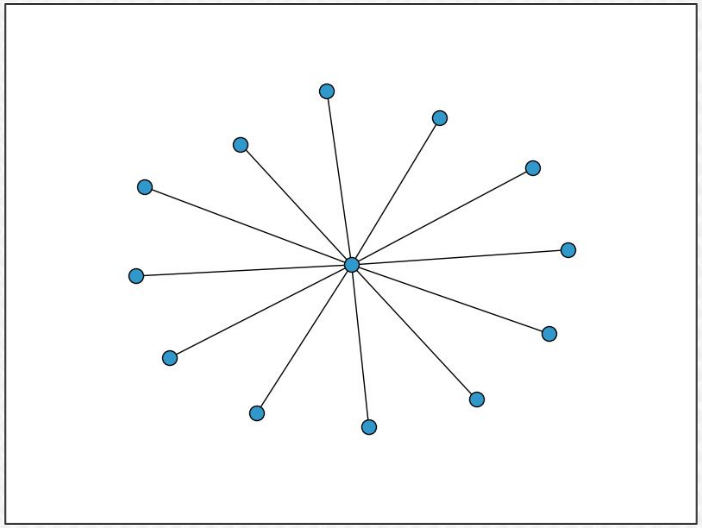
Balanced Tree (Hierarchy)



Bipartite Graph



Complete Graph



Star Graph

Describing Graphs by Structure

Traversal is about following *paths* between vertices

$$p = \langle v_i, \ldots, v_j \rangle \quad (v_{k-1}, v_k \in E)$$

a path from `i` to k

`Length(p)` = # of nodes in path
`Paths(i,j)` = set of paths from `i` to `j`

$$L(i,j) = \min(\{length(p) \mid p \in Paths(i,j)\})$$

Shortest (unweighted) path length

Paths in a Network

Describing Graphs by Path

Average Shortest Path

$$\ell = \frac{1}{n} \sum_{i \in V} \ell(i)$$

`Diameter(G)`: the "longest shortest path"

Characteristic Path Length

$$CPL = median(\{\ell(i) | i \in V\})$$

Classes of Graph Algorithms

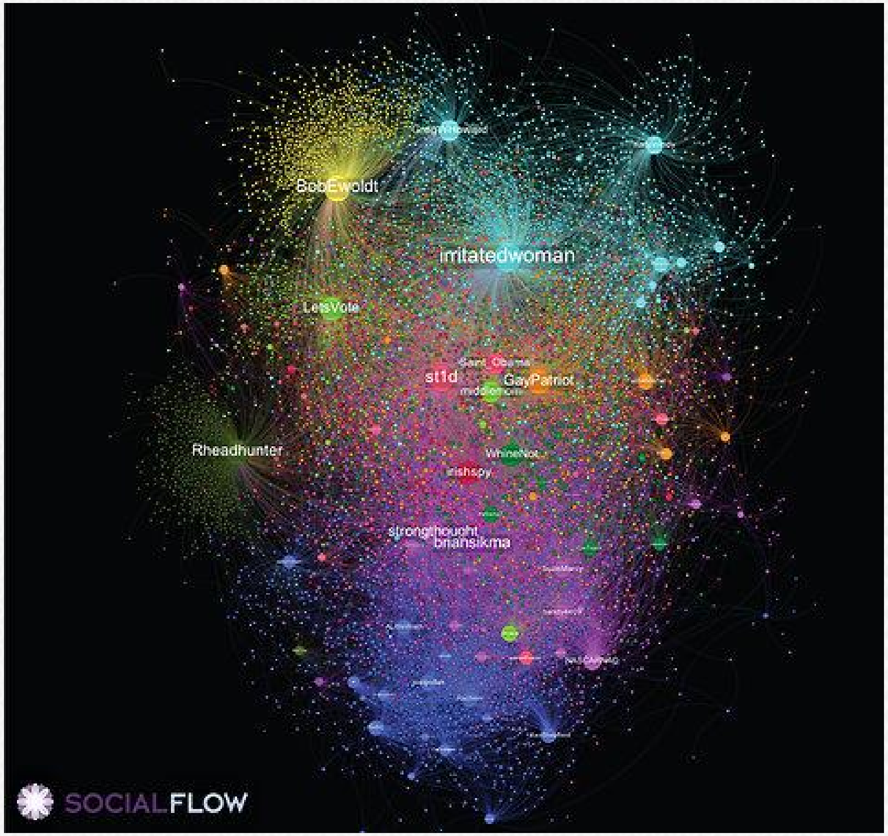
Generally speaking CS Algorithms are designed to solve the classes of Math problems, but we can further categorize them into the following:

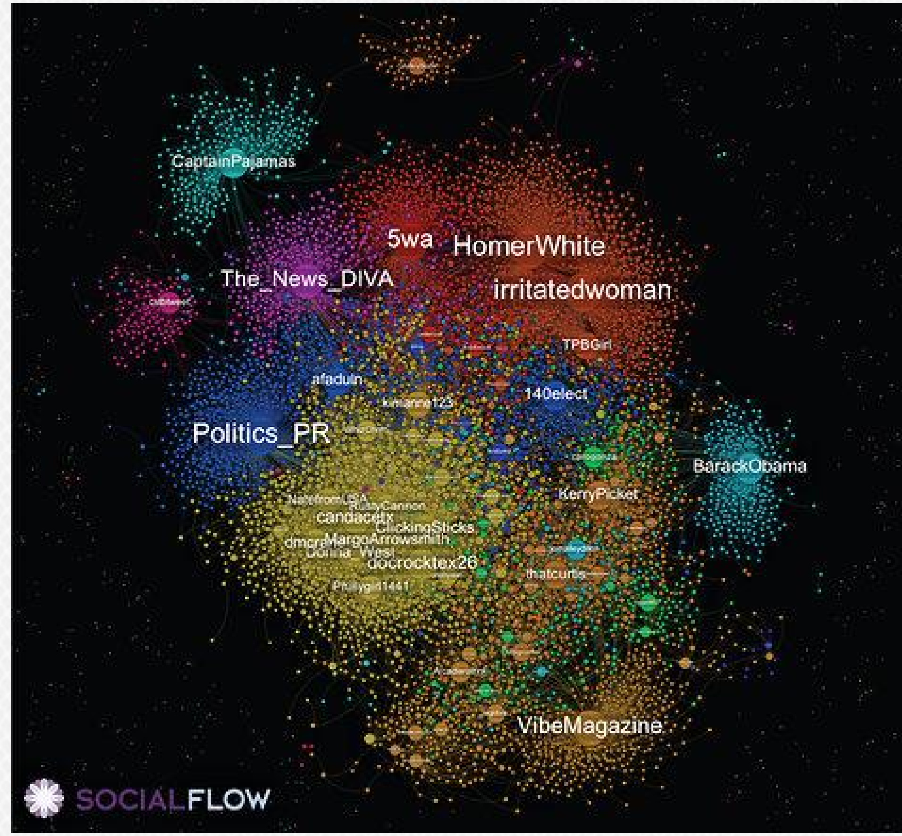
1. **Traversal** (flow, shortest distance)
2. **Search** (optimal node location)
3. **Subgraphing** (find minimum weighted spanning tree)
4. **Clustering** (group neighbors of nodes)

For Reference

Bellman-Ford Algorithm | Dijkstra's Algorithm
Ford-Fulkerson Algorithm | Kruskai's Algorithm
Nearest neighbor | Depth-First and Breadth-First

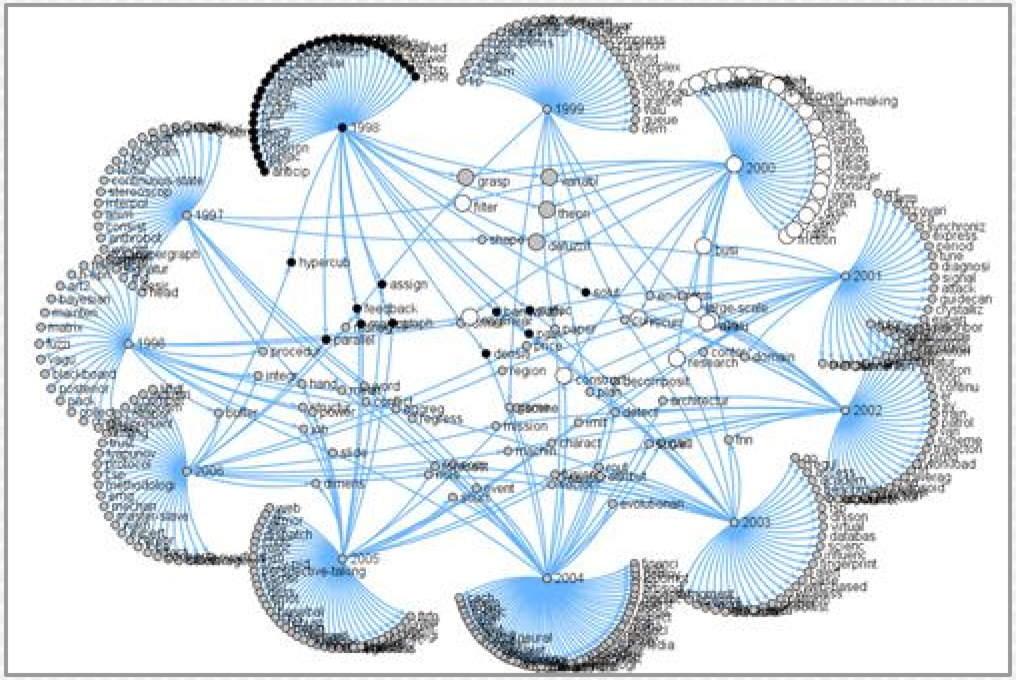
So why are Graphs important?





Ryan vs. Biden Debate (Twitter Reaction)

http://thecaucus.blogs.nytimes.com/2012/10/16/who-won-presidential-debate-on-twitter/?_r=0



Topics shifting over time

http://informationandvisualization.de/blog/graphbased-visualization-topic-shifts

Data Loading Benchmark

PEARSON

Pearson provides free online education through its OpenClass platform. OpenClass is currently in beta with adoption by ~7000 institutions. To meet the expected growth beyond beta, it is necessary to build the platform on a scalable database system. Moreover, it is important to build the platform on a database system that can support advanced algorithms beyond simple get/put-semantics. The latter was demonstrated via the initial prototype. To the former, a simulation of a worldwide education environment was created in Titan to alleviate Pearson's scalability concerns.

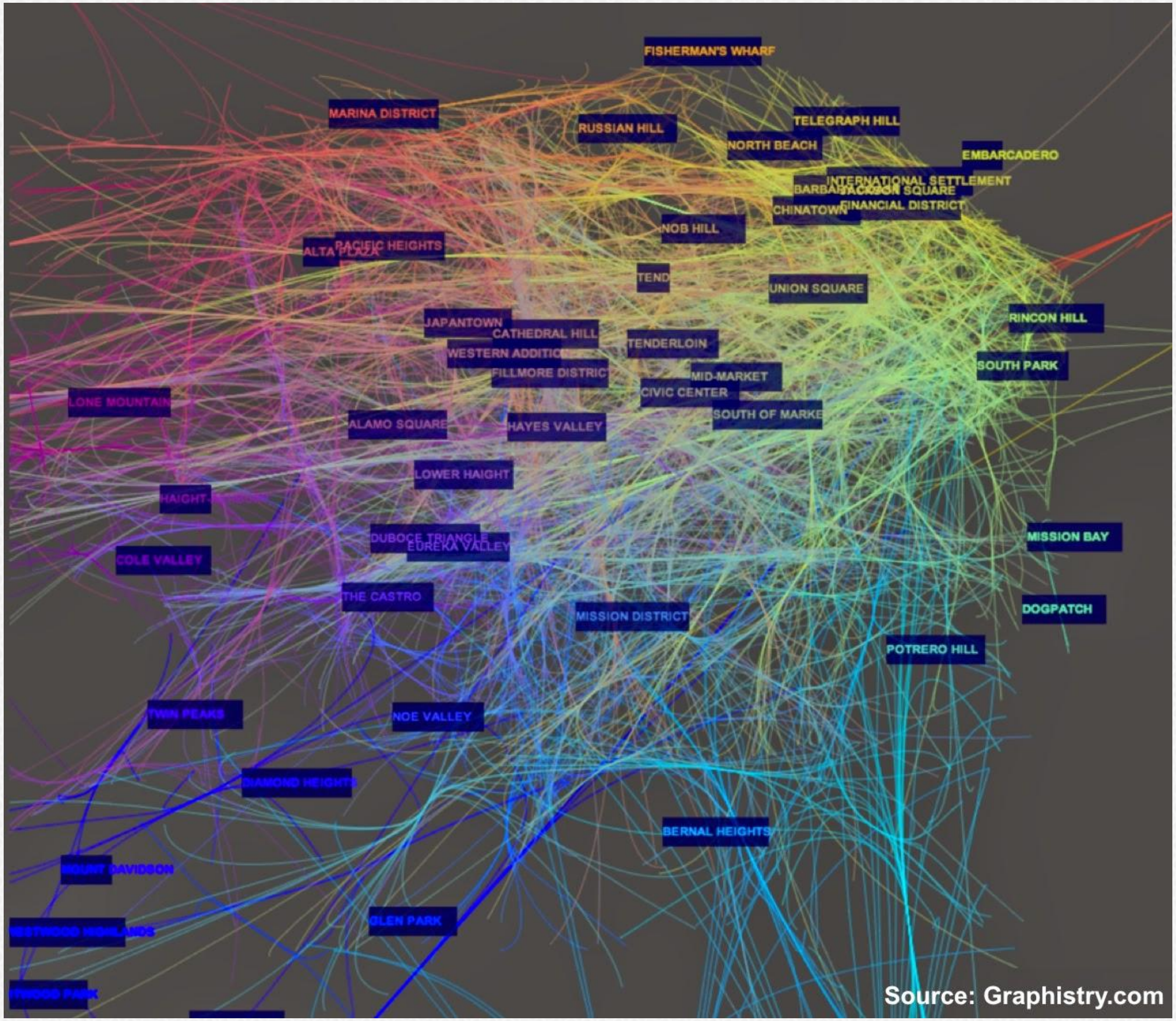
OpenClass™

The simulated world was a graph containing 6.24 billion vertices and 121 billion edges. The edges represent students enrolled in courses, people discussing content, content referencing concepts, teachers teaching courses, material contained in activity streams, universities offering courses, and so forth. Various techniques were leveraged to ensure that the generated data was consistent with a real-world instance. For example, people names were generated from sampling the cross product of the first and last names in the US Census Bureau dataset. Gaussian distributions were applied to determine how many courses a student should be enrolled in (mean of 8) and how many courses a teacher should teach (mean of 4). The course names and descriptions were drawn from the raw MIT OpenCourseWare data dumps. Course names were appended with tokens such as "101," "1B," "Advanced," etc. in order to increase the diversity of the offerings. Student comments in discussions were sampled snippets of text from the electronic books provided by Project Gutenberg. University names were generated from publicly available city name and location datasets in the CommonHubData project. Finally, concepts were linked to materials using OpenCalais. The final raw education dataset was 10 terabytes in size.

Number of students	3.47 billion
Number of teachers	183 million
Number of courses	788 million
Number of universities	1.2 million
Number of concepts	9 thousand
Number of educational artifacts	~1.5 billion
Total number of vertices	**6.24 billion**
Total number of edges	**121 billion**

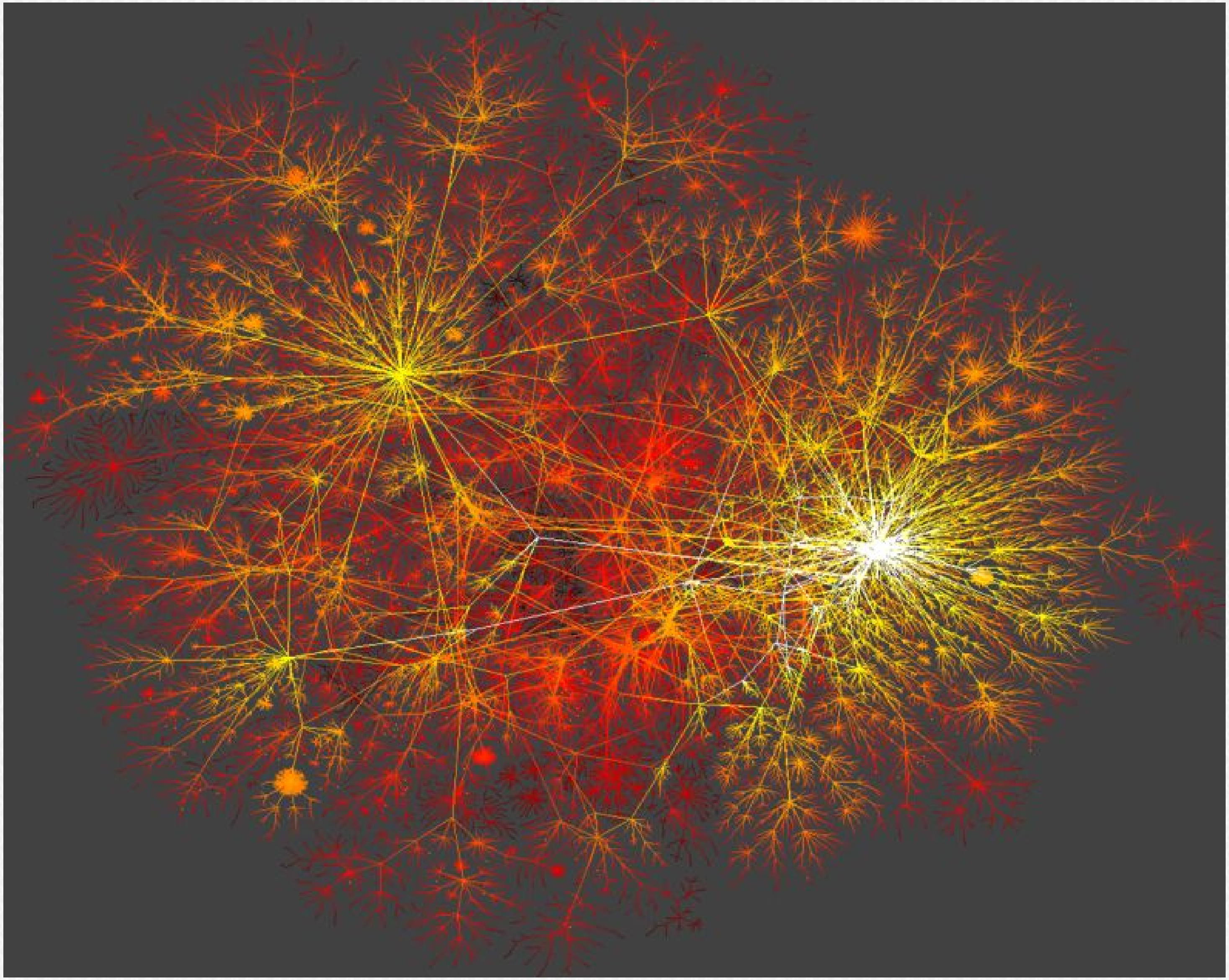
Pearson OpenClass Graph

http://thinkaurelius.com/2013/05/13/educating-the-planet-with-pearson/



Graph Analysis for Big Data (Uber Trips in San Francisco)

http://radar.oreilly.com/2014/07/there-are-many-use-cases-for-graph-databases-and-analytics.html



Information Flows

http://web.math.princeton.edu/math_alive/5/Lab1/Networks.html

Why Graphs?

1. Graphs are abstractions of real life
2. Represent information flows that exist
3. Explicitly demonstrate relationships
4. Enable computations across large datasets
5. Allow us to compute locally to areas of interest with small traversals
6. Because everyone else is doing it (PageRank, SocialGraph)

Why are graphs useful for analytics?

Humans can understand and interpret network structures, leading to insight.

Performance boost to some machine learning algorithms

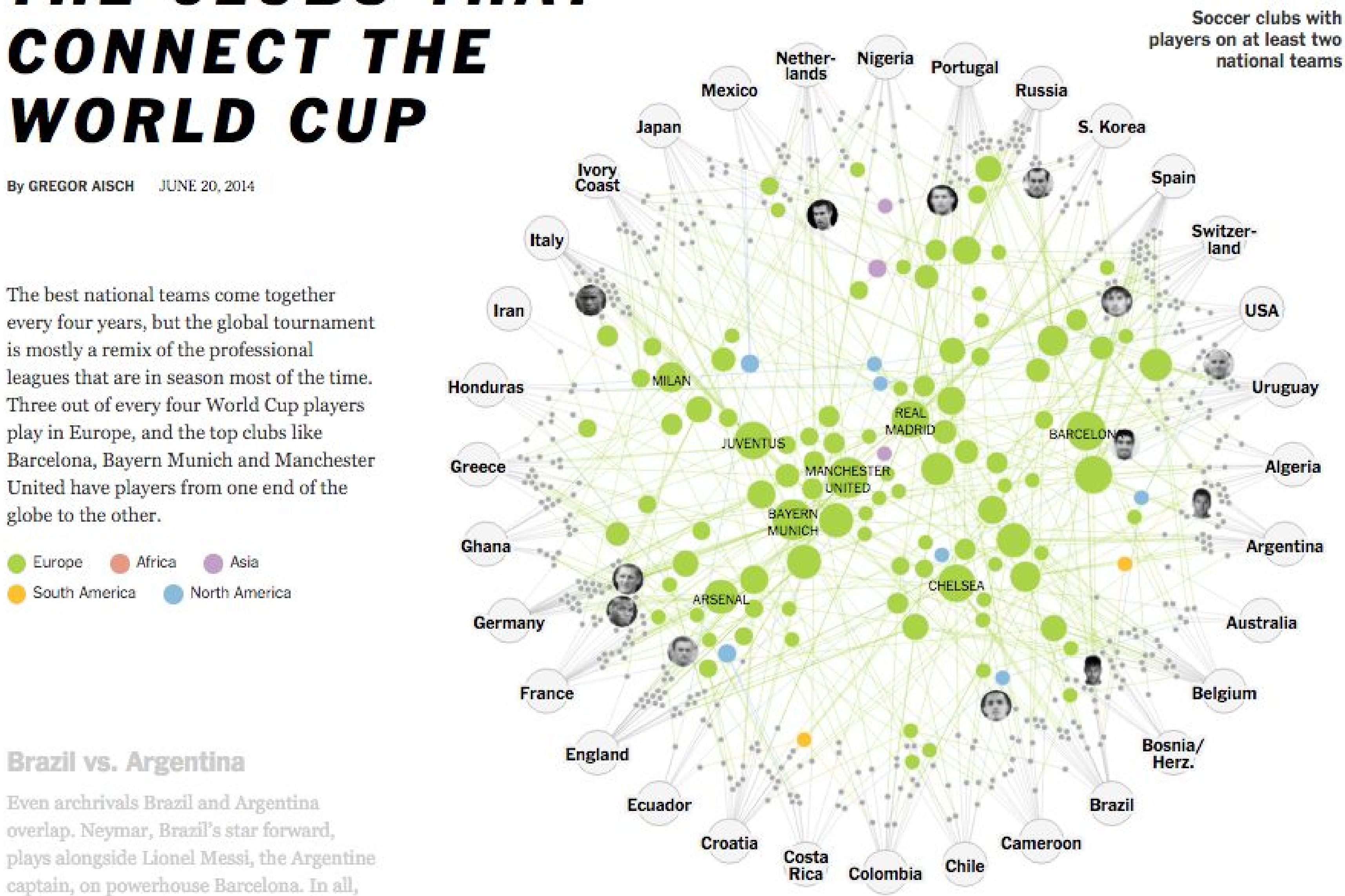
Brazil 2014

SHARE

THE CLUBS THAT CONNECT THE WORLD CUP

By GREGOR AISCH JUNE 20, 2014

The best national teams come together every four years, but the global tournament is mostly a remix of the professional leagues that are in season most of the time. Three out of every four World Cup players play in Europe, and the top clubs like Barcelona, Bayern Munich and Manchester United have players from one end of the globe to the other.



Brazil vs. Argentina

Even archrivals Brazil and Argentina overlap. Neymar, Brazil's star forward, plays alongside Lionel Messi, the Argentine captain, on powerhouse Barcelona. In all, eight Brazilians and 12 Argentines play together on European club teams.

http://nyti.ms/1Yd1BPT

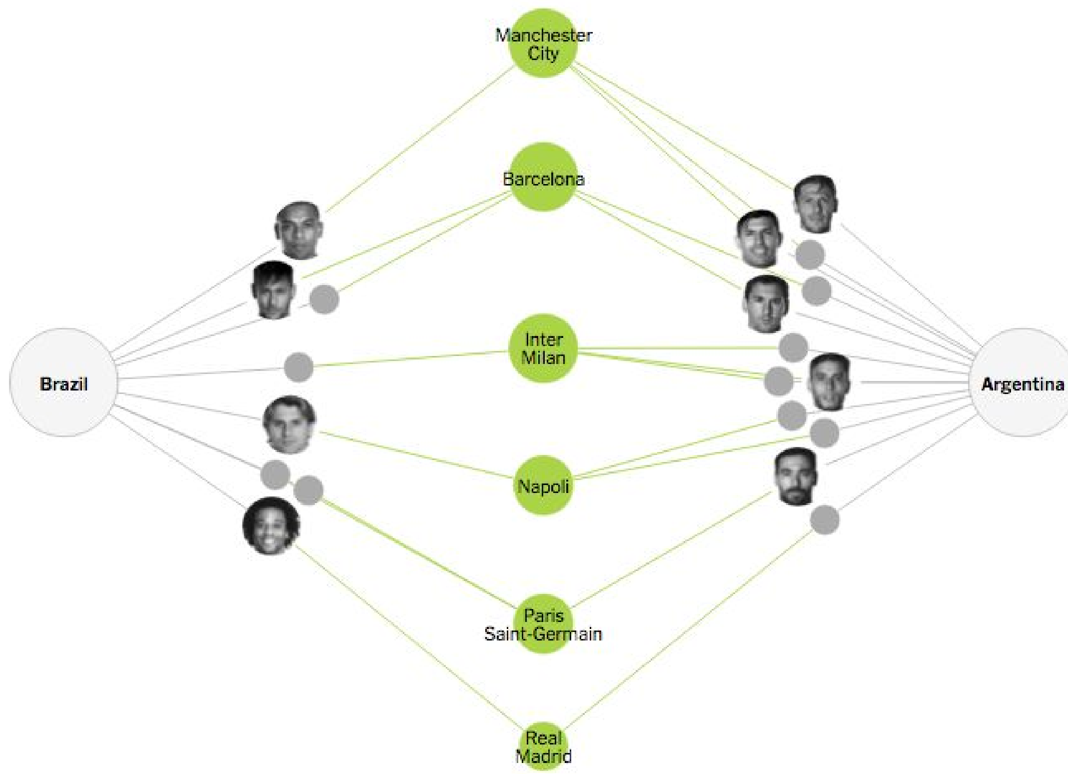
most of the time. Three out of every four World Cup players play in Europe, and the top clubs like Barcelona, Bayern Munich and Manchester United have players from one end of the globe to the other.

Europe Africa Asia
South America North America

Soccer club connections between the national teams of Argentina and Brazil

Brazil vs. Argentina

Even archrivals Brazil and Argentina overlap. Neymar, Brazil's star forward, plays alongside Lionel Messi, the Argentine captain, on powerhouse Barcelona. In all, eight Brazilians and 12 Argentines play together on European club teams.



The European Connection

Three out of every four World Cup players play on a European professional team. Bayern Munich and Manchester United top the list, each with 14 players on World Cup teams. More than half of the European club players in the World Cup play in one of the four strongest primary

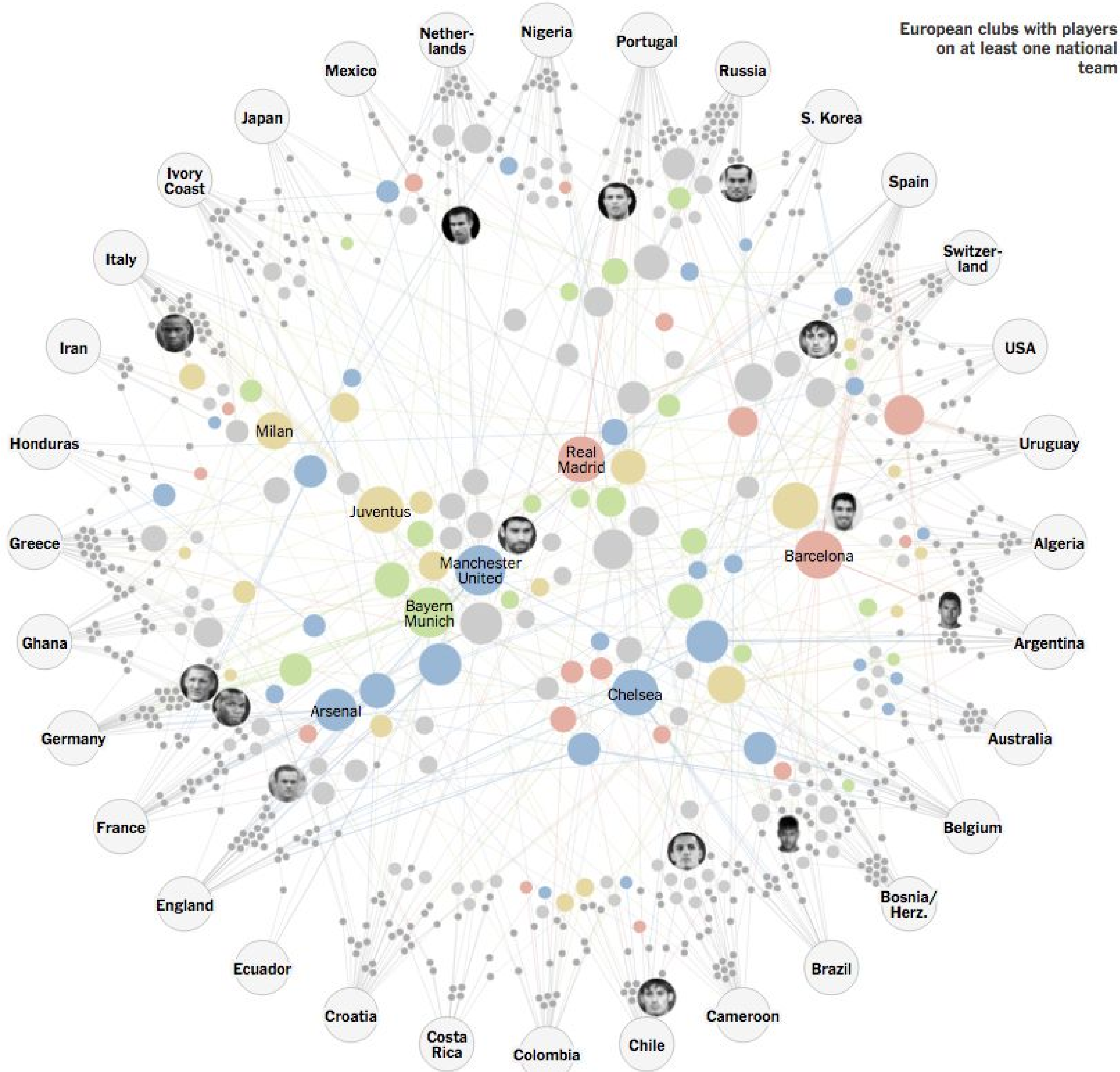
http://nyti.ms/1Yd1BPT

Argentines play together on European club teams.

The European Connection

Three out of every four World Cup players play on a European professional team. Bayern Munich and Manchester United top the list, each with 14 players on World Cup teams. More than half of the European club players in the World Cup play in one of the four strongest primary leagues (in England, Spain, Germany and Italy).

- English Premier League
- Spanish La Liga
- German Bundesliga
- Italian Serie A
- Other Leagues



The Rest of the World

Meanwhile, leagues outside Europe account for only 24 percent of the World Cup players, and only a handful of those teams supply players for more than one World Cup team. M.L.S. teams account for only 3 percent of the World Cup players.

Africa Asia South America

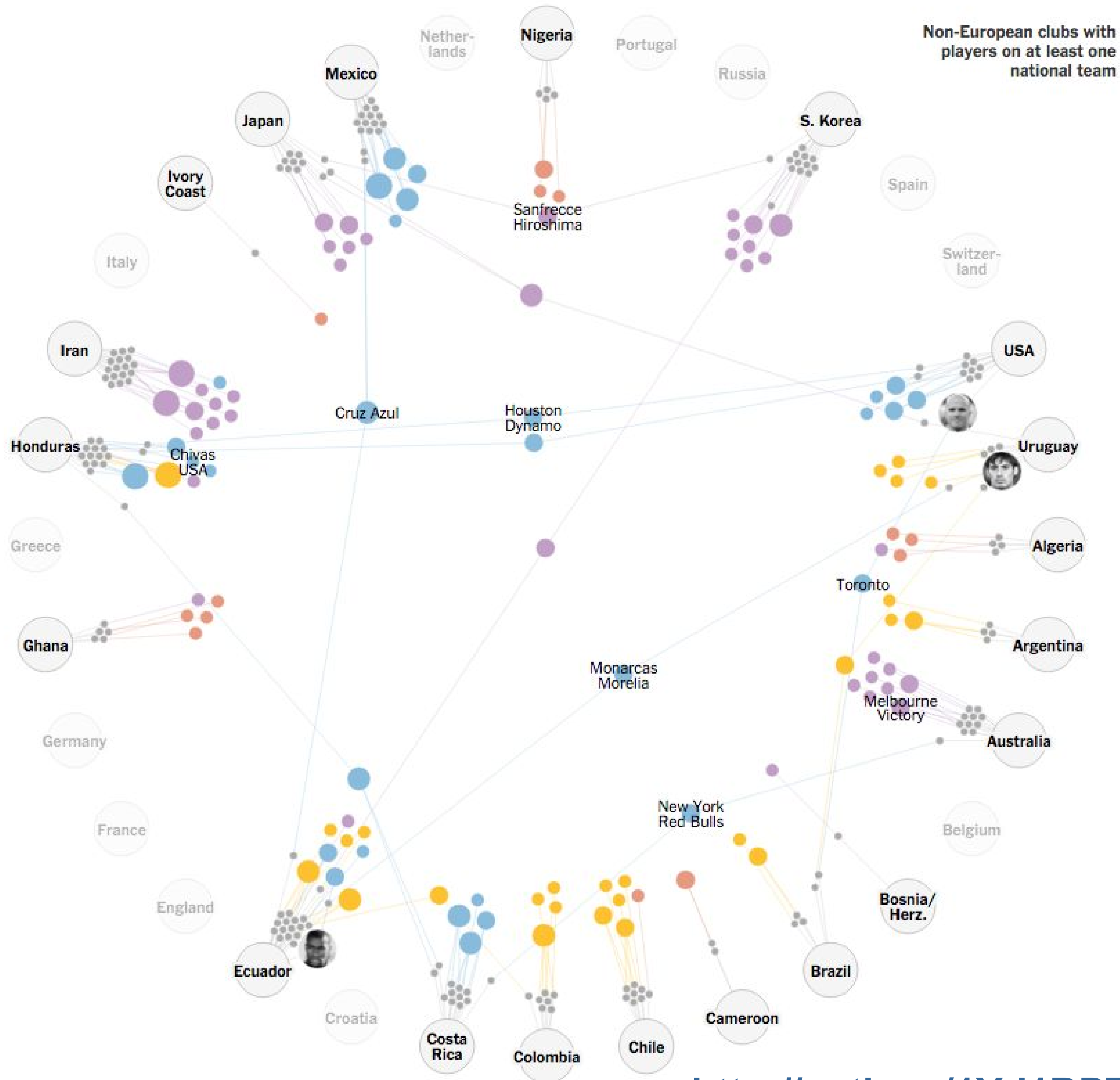
http://nyti.ms/1Yd1BPT

Spanish La Liga
German Bundesliga
Italian Serie A
Other Leagues

The Rest of the World

Meanwhile, leagues outside Europe account for only 24 percent of the World Cup players, and only a handful of those teams supply players for more than one World Cup team. M.L.S. teams account for only 3 percent of the World Cup players.

Africa · Asia · South America · North America



Find a Team

Why not look around for yourself?

Filter World Cup teams by:

Region	All
Group	All
Stage	All

Filter clubs by:

Region	All

http://nyti.ms/1Yd1BPT

Humans can understand and interpret network structures, leading to insight.

Performance boost to some machine learning algorithms

Machine Learning using Graphs

- Machine Learning is *iterative* but iteration can also be seen as *traversal*.

Machine Learning using Graphs

- Machine Learning is *iterative* but iteration can also be seen as *traversal*.
- Machine Learning requires many instances with which to fit a model to make predictions.

Machine Learning using Graphs

- Machine Learning is *iterative* but iteration can also be seen as *traversal*.
- Machine Learning requires many instances with which to fit a model to make predictions.
- Important analyses are graph algorithms: clusters, influence propagation, centrality.

Machine Learning using Graphs

- Machine Learning is *iterative* but iteration can also be seen as *traversal*.
- Machine Learning requires many instances with which to fit a model to make predictions.
- Important analyses are graph algorithms: clusters, influence propagation, centrality.
- Performance benefits on sparse data

Machine Learning using Graphs

- Machine Learning is *iterative* but iteration can also be seen as *traversal*.
- Many domains have structures already modeled as graphs (health records, finance)
- Important analyses are graph algorithms: clusters, influence propagation, centrality.
- Performance benefits on sparse data
- More understandable implementation

Iterative PageRank in Python

```
def pageRank(G, s = .85, maxerr = .001):
    n = G.shape[0]

    # transform G into markov matrix M
    M = csc_matrix(G,dtype=np.float)
    rsums = np.array(M.sum(1))[:,0]
    ri, ci = M.nonzero()
    M.data /= rsums[ri]
    sink = rsums==0  # bool array of sink states

    # Compute pagerank r until we converge
    ro, r = np.zeros(n), np.ones(n)
    while np.sum(np.abs(r-ro)) > maxerr:
        ro = r.copy()
        for i in xrange(0,n):
            Ii = np.array(M[:,i].todense())[:,0] # inlinks of state i
            Si = sink / float(n)                 # account for sink states
            Ti = np.ones(n) / float(n)           # account for teleportation
            r[i] = ro.dot( Ii*s + Si*s + Ti*(1-s) )

    return r/sum(r) # return normalized pagerank
```

Graph-Based PageRank in Gremlin

```
pagerank = [:].withDefault{0}
size = uris.size();
uris.each{
    count = it.outE.count();
    if(count == 0 || rand.nextDouble() > 0.85) {
        rank = pagerank[it]
        uris.each {
            pagerank[it] = pagerank[it] / uris.size()
        }
    }
    rank = pagerank[it] / it.outE.count();
    it.out.each{
        pagerank[it] = pagerank[it] + rank;
    }
}
```

Visualization

Humans can understand and interpret network structures, leading to insight.

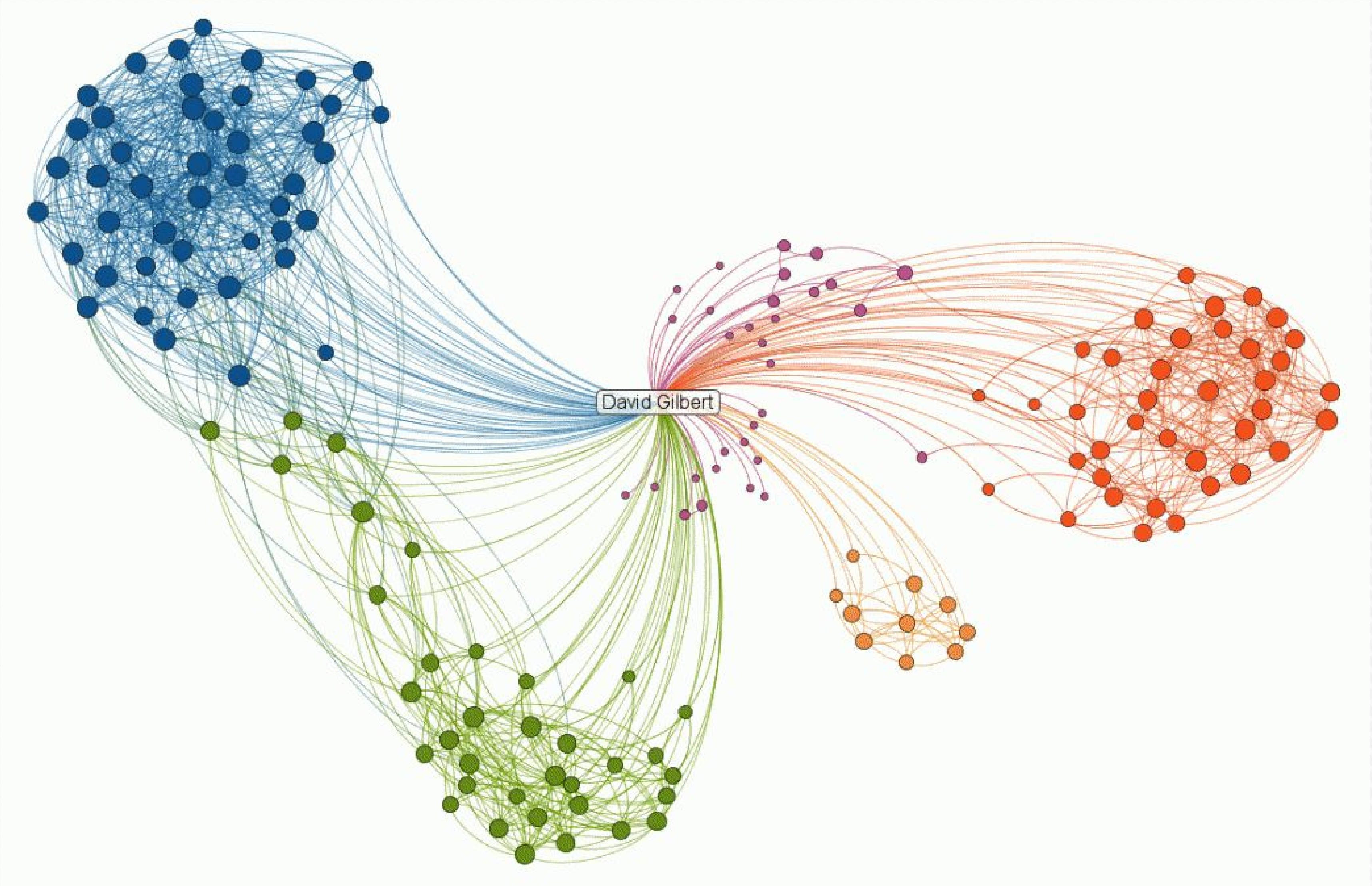
Performance boost to some machine learning algorithms

High Performance Computation

Classes of Graph Analyses

- **Existence**: Does there exist [a path, a vertex, a set] within [constraints]?
- **Construction**: Given a set of [paths, vertices] is a [constraint] graph construction possible?
- **Enumeration**: How many [vertices, edges] exist with [constraints], is it possible to list them?
- **Optimization**: Given several [paths, etc.] is one the best?

Social Networks



… are graphs!

http://randomwire.com/linkedin-inmaps-visualises-professional-connections/

A **social network** is a data structure whose nodes are composed of *actors* (proper nouns except places) that transmit information to each other according to their *relationships* (links) with other *actors*.

Semantic definitions of both actors and relationships are illustrative:

> **Actor:** person, organization, place, role
> **Relationship:** friends, acquaintance, penpal, correspondent

Social networks are *complex* - they have non-trivial topological features that do not occur in simple networks.

Almost any system humans participate and communicate in can be modeled as a social network (hence rich semantic relevance)

Attributes of a Social Network

- Scale Free Networks

  http://en.wikipedia.org/wiki/Scale-free_network

  - Degree distribution follows a power law
  - Significant topological features (not random)
  - Commonness of vertices with a degree that greatly exceeds the average degree (“hubs”) which serve some purpose

- Small World Networks

  http://en.wikipedia.org/wiki/Small-world_network

  - Most nodes aren’t neighbors but can be reached quickly
  - Typical distance between two nodes grows proportionally to the logarithm of the order of the network.
  - Exhibits many specific clusters

Degree Distribution

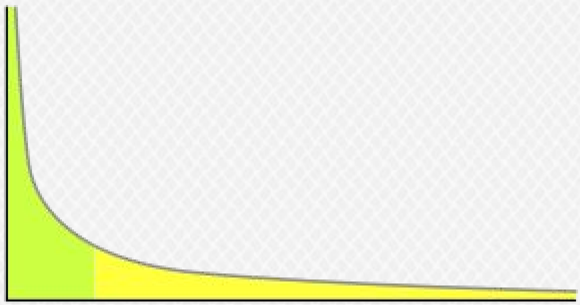
- We’ve looked so far at per node properties (degree, etc) and averaging them gives us some information.
- Instead, let’s look at the entire distribution
-

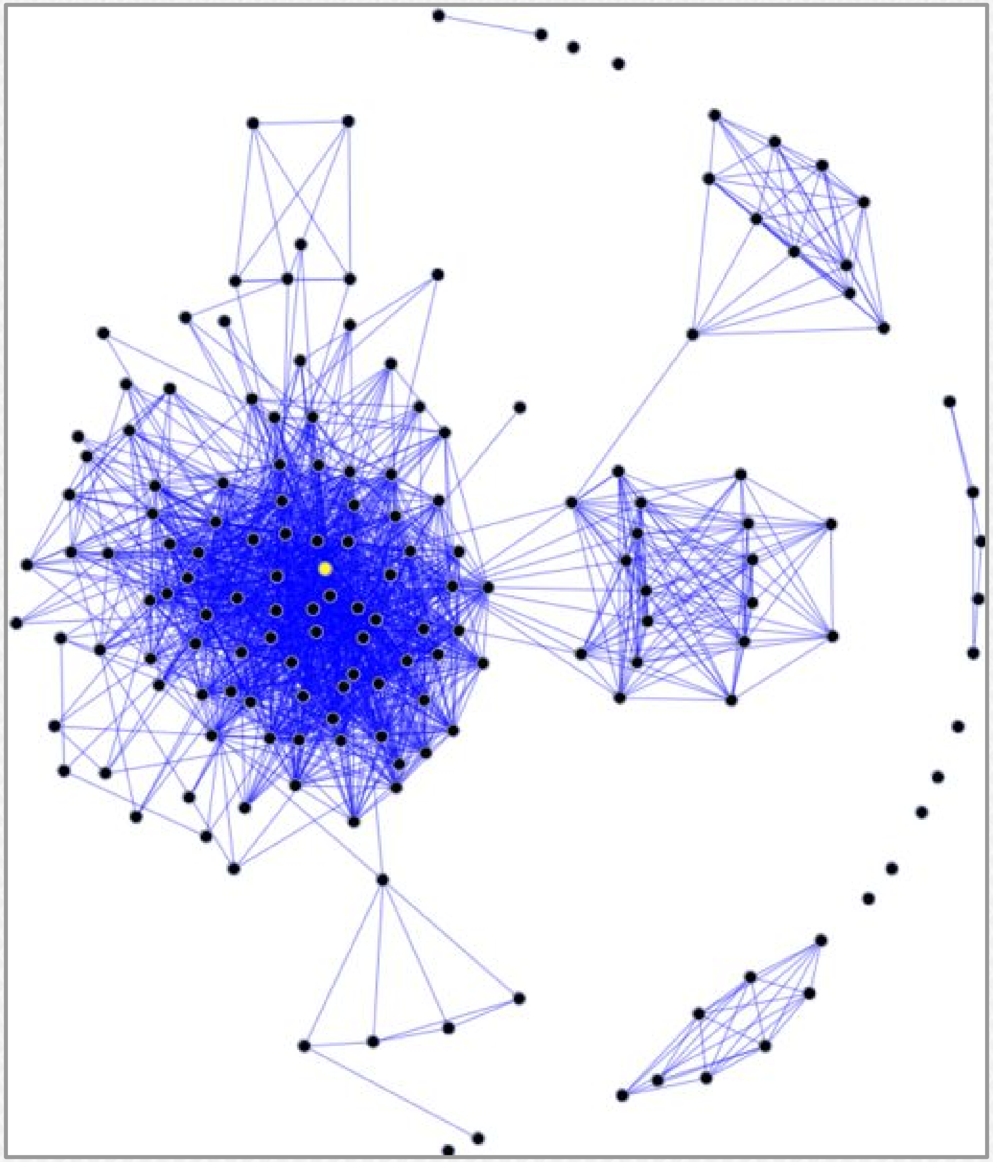
Degree Distribution:

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k},$$

Power Law distribution:

$$p_k \propto k^{-\gamma} \quad \gamma > 1$$



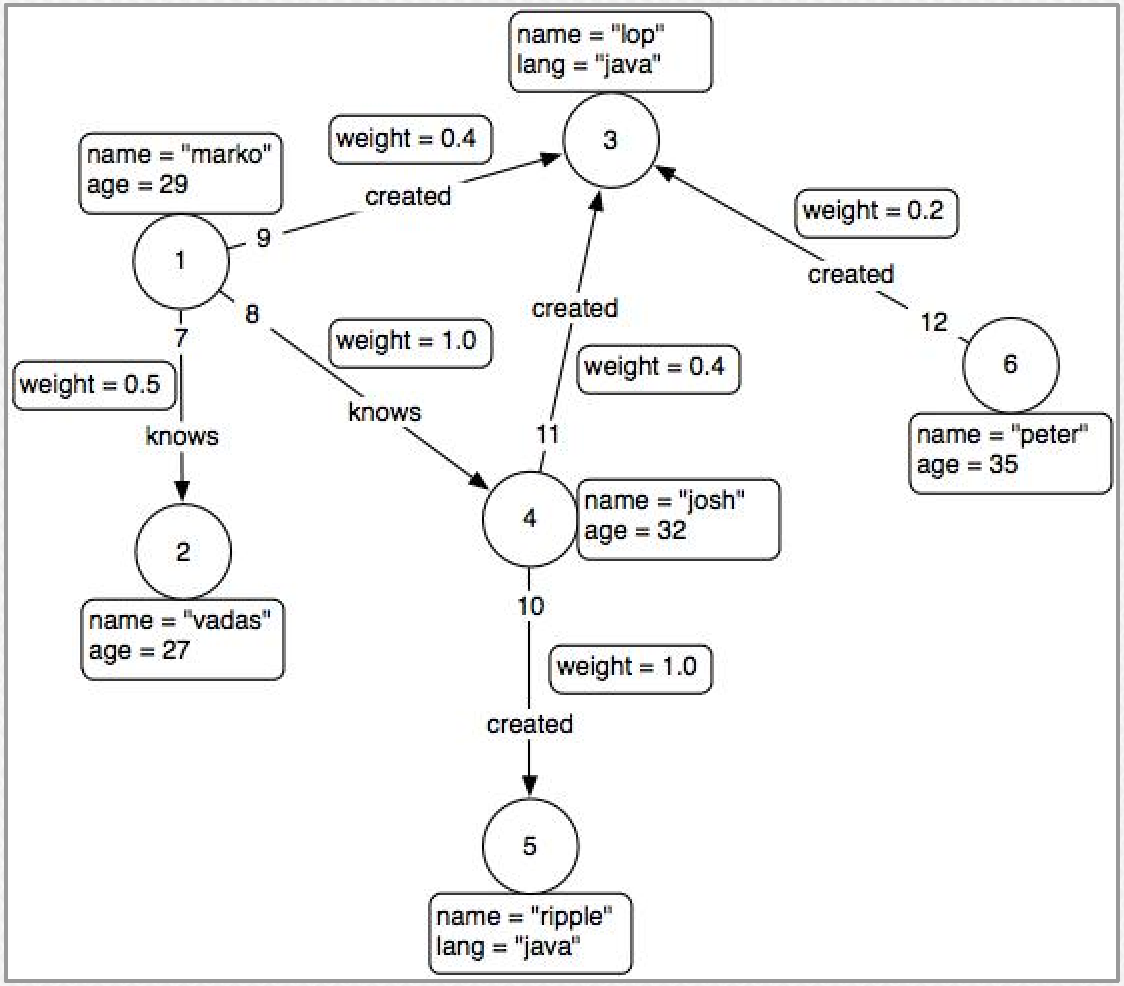


Network Topology

http://filmword.blogspot.com/2010/04/emerging-brain.html

Graphs as Data

Where do graphs that we can analyze come from?



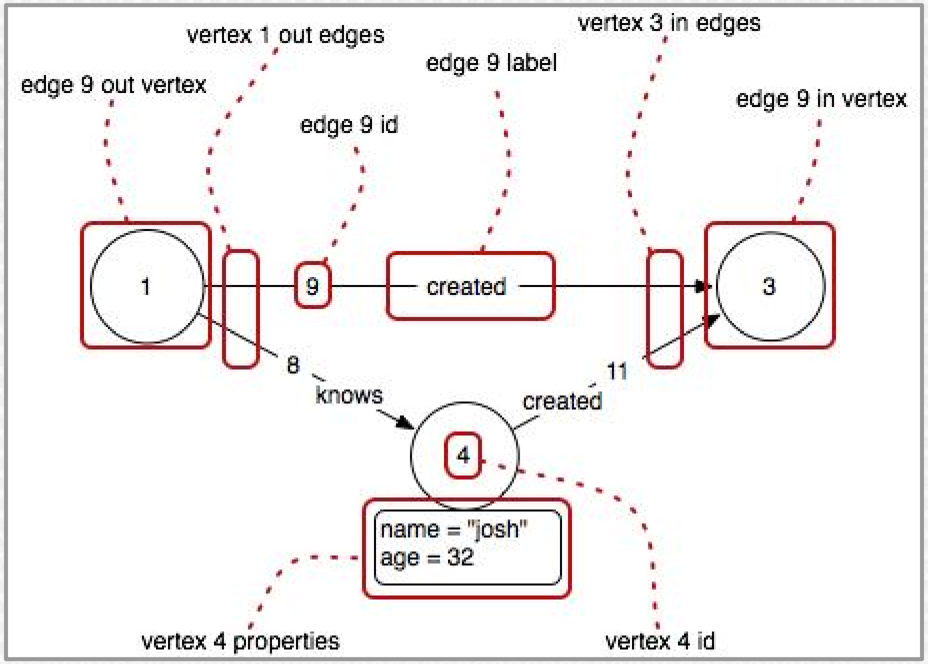
Graphs contain semantically relevant information - “Property Graph”

https://github.com/tinkerpop/blueprints/wiki/Property-Graph-Model

Property Graph Model

The primary data model for Graphs, containing these elements:

1. a set of vertices
   - each vertex has a unique identifier.
   - each vertex has a set of outgoing edges.
   - each vertex has a set of incoming edges.
   - each vertex has a collection of properties defined by a map from key to value.
2. a set of edges
   - each edge has a unique identifier.
   - each edge has an outgoing tail vertex.
   - each edge has an incoming head vertex.
   - each edge has a label that denotes the type of relationship between its two vertices.
   - each edge has a collection of properties defined by a map from key to value.

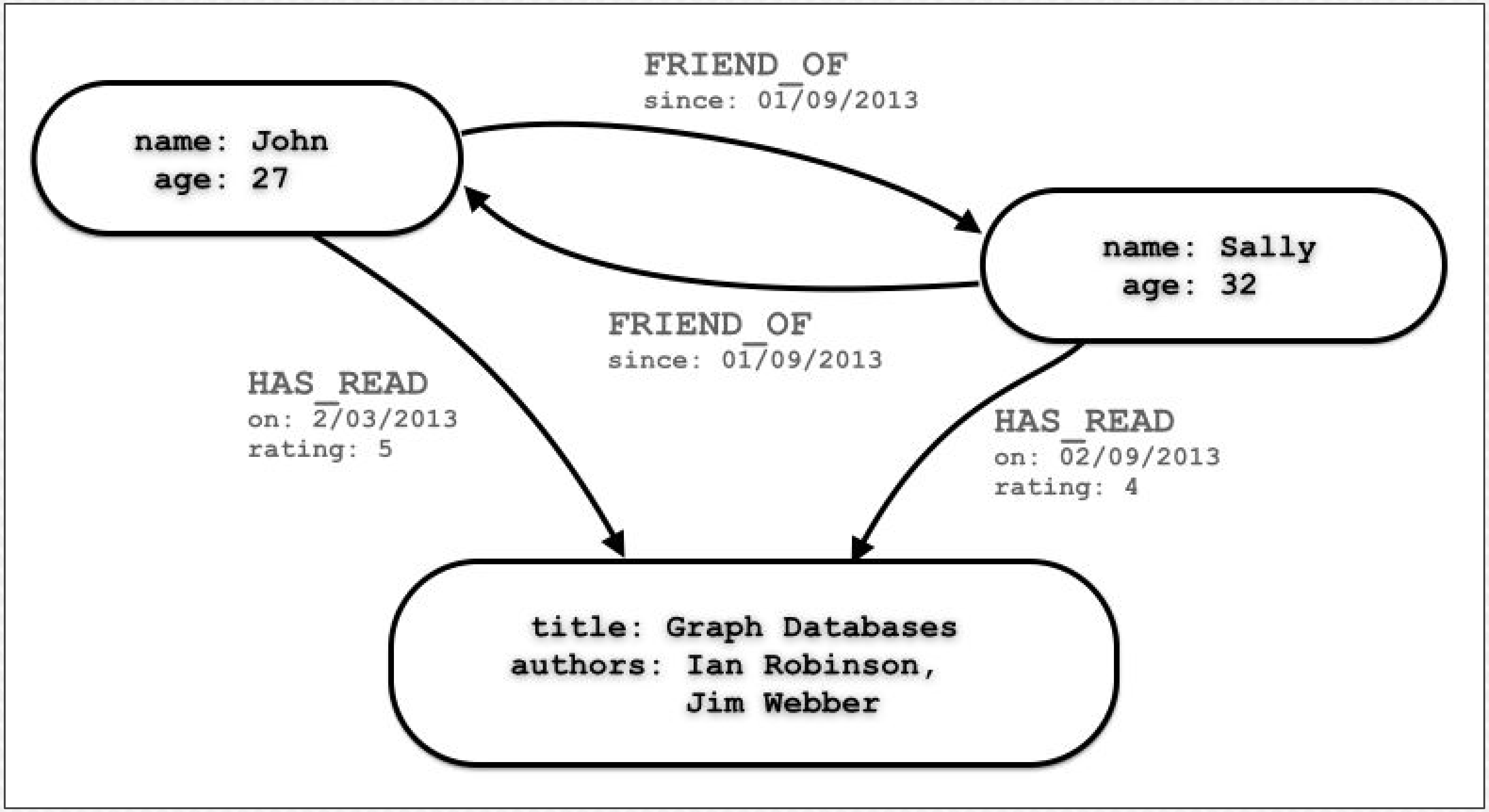


Graph: An object that contains vertices and edges.

Element: An object that can have any number of key/value pairs associated with it (i.e. properties)

Vertex: An object that has incoming and outgoing edges.

Edge: An object that has a tail and head vertex.



Modeling property graphs with labels, relationships, and properties.

http://neo4j.com/developer/guide-data-modeling/

Getting out of Memory

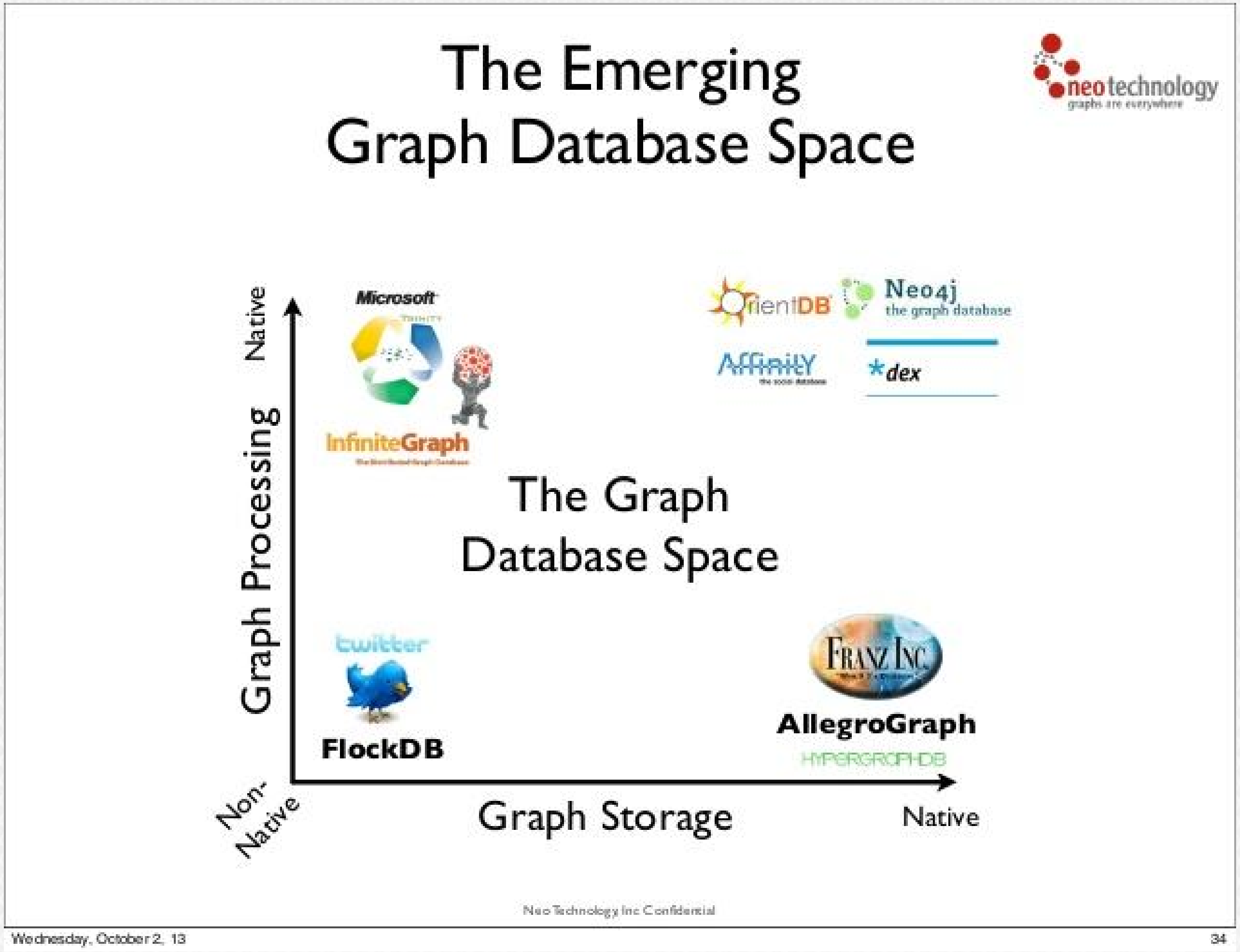
<xml />
GraphML

{json}
GraphSon

NetworkX

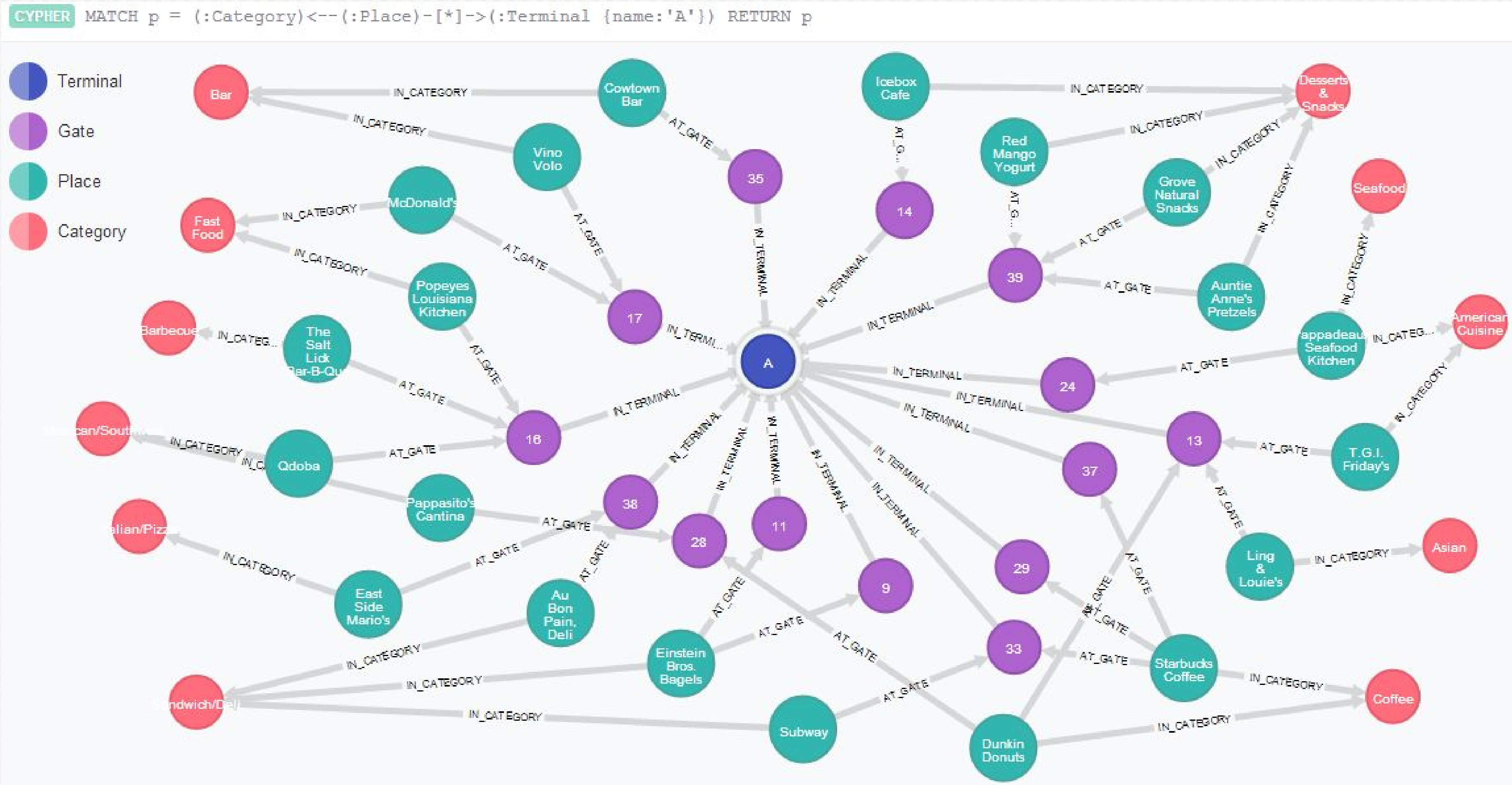
Gephi

File Based Serialization

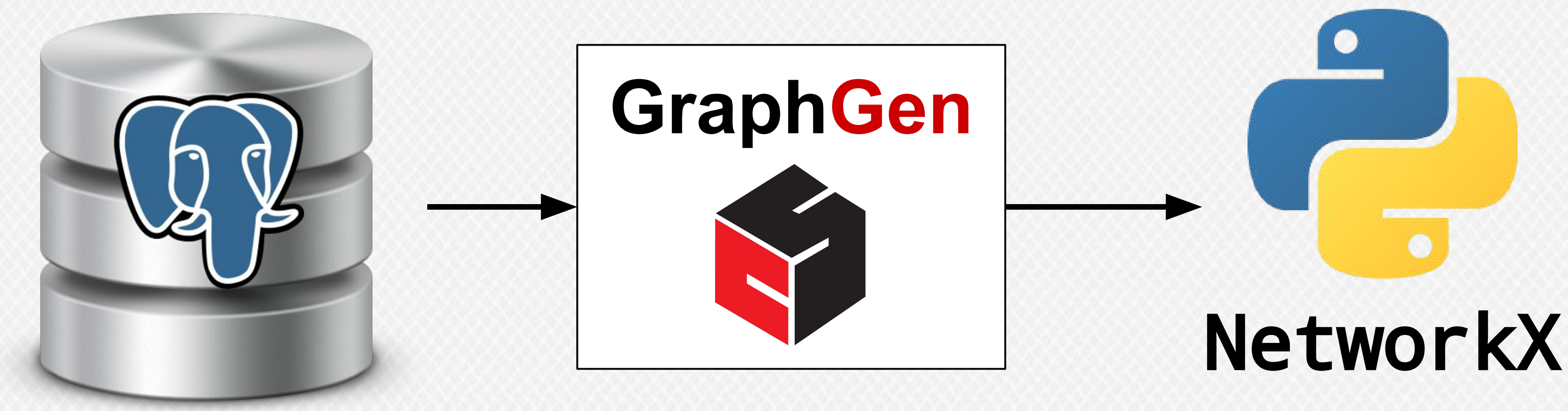


Graph Databases

http://graphdatabases.com/



Neo4j: Querying with Cypher and a visual interface.



Relational Data: Use GraphGen

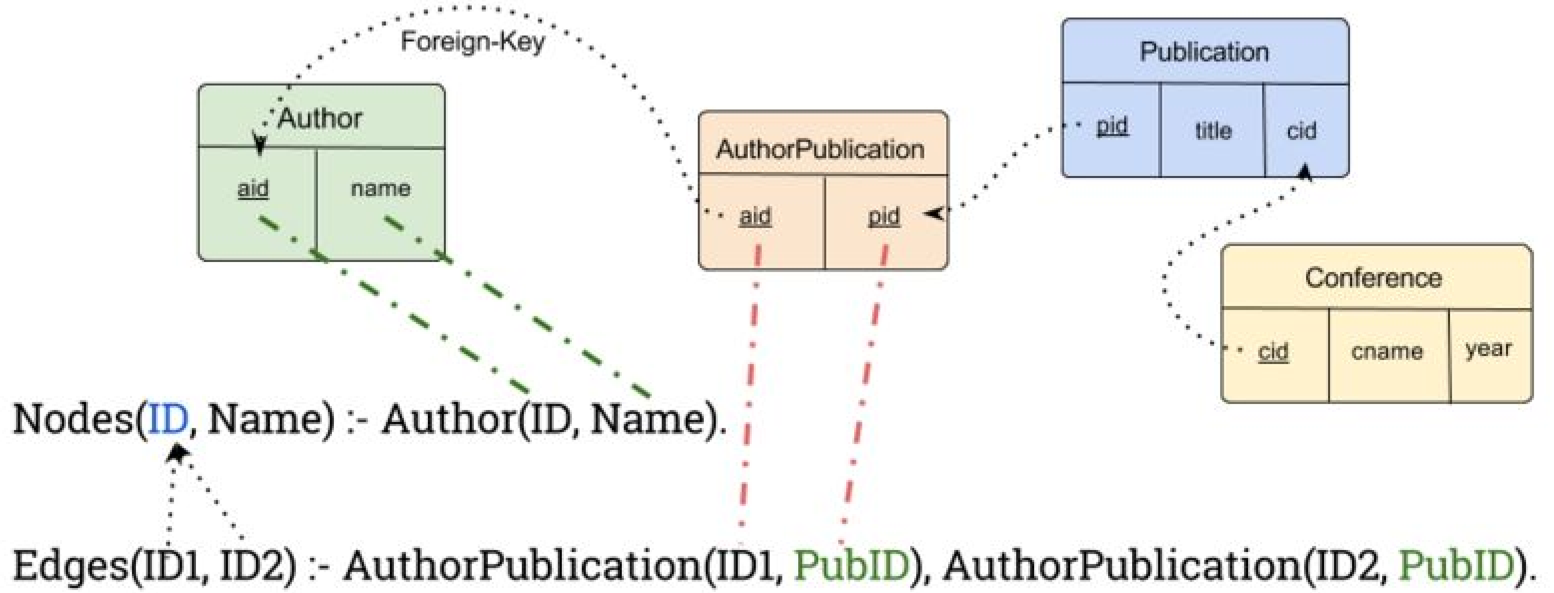
http://konstantinosx.github.io/graphgen-project/

Extract a graph where authors are connected to each other if they've published a paper together:

```
Nodes(ID, Name) :- Author(ID, Name).

Edges(ID1, ID2) :- AuthorPublication(ID1, PubID), AuthorPublication(ID2, PubID).
```

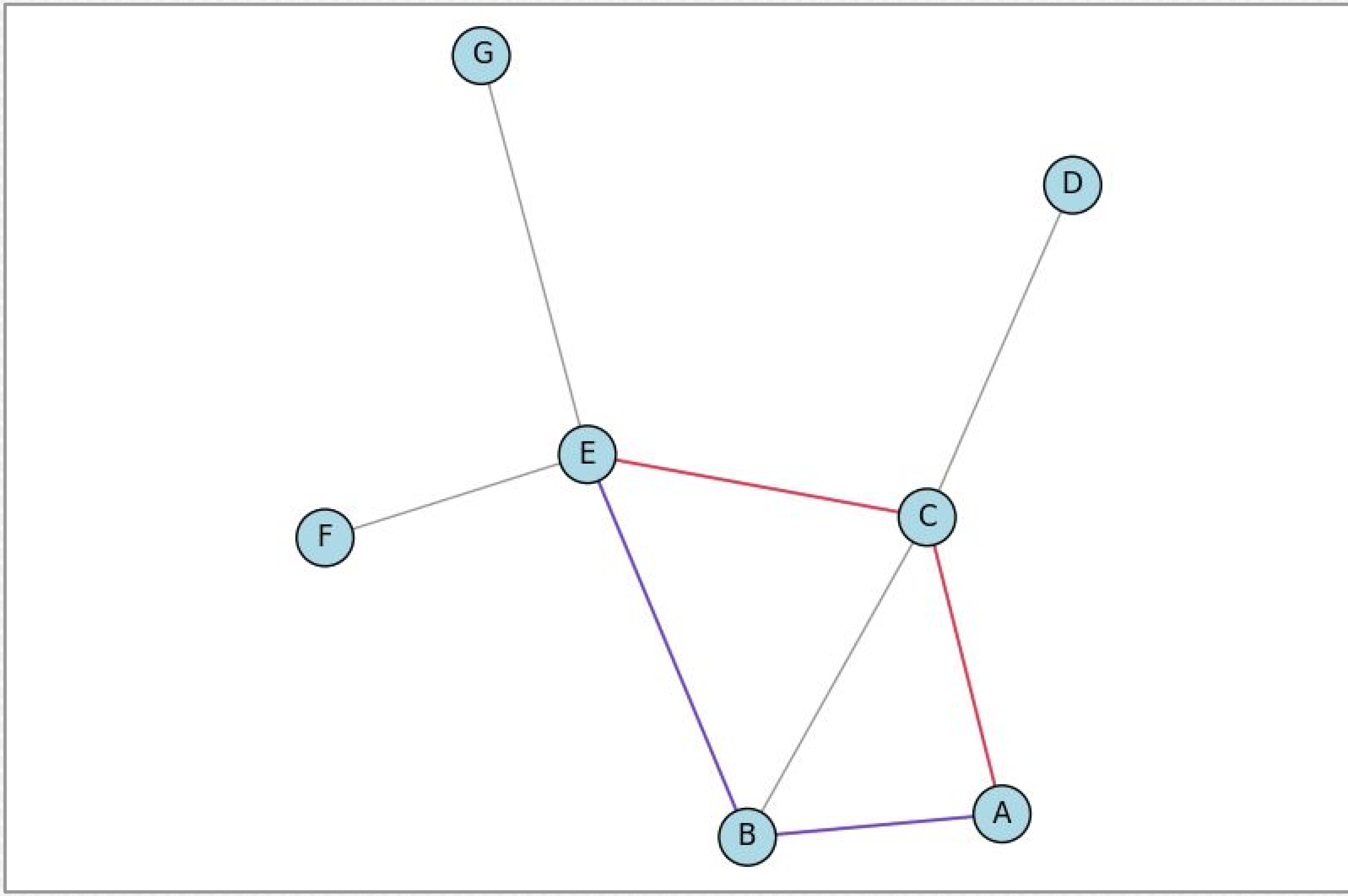
Let's take a look at how this query is formulated!



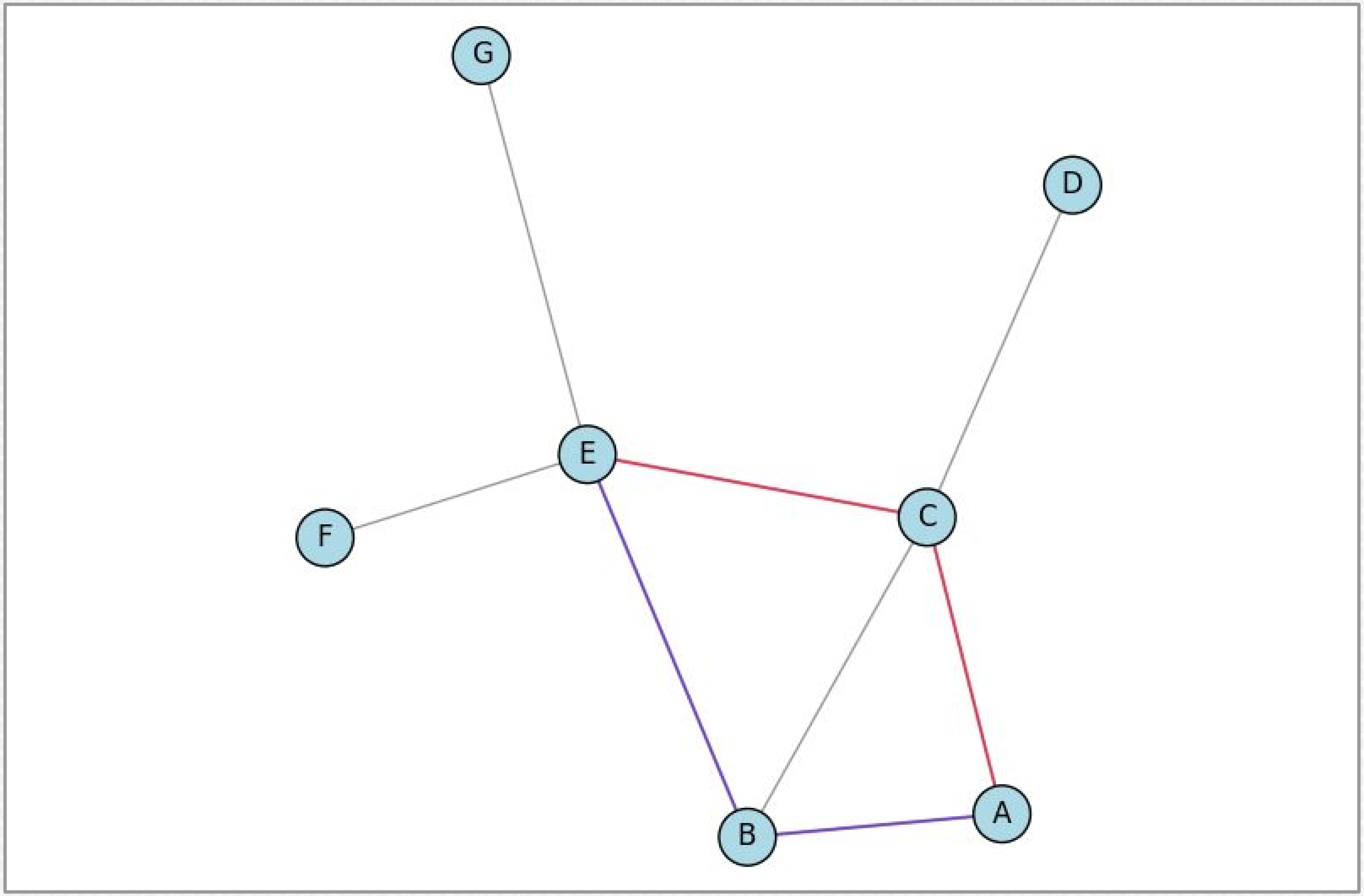
Relational Data: Use GraphGen

http://konstantinosx.github.io/graphgen-project/

Graph Analytics



Sample graph - note the shortest paths from D to F and A to E.



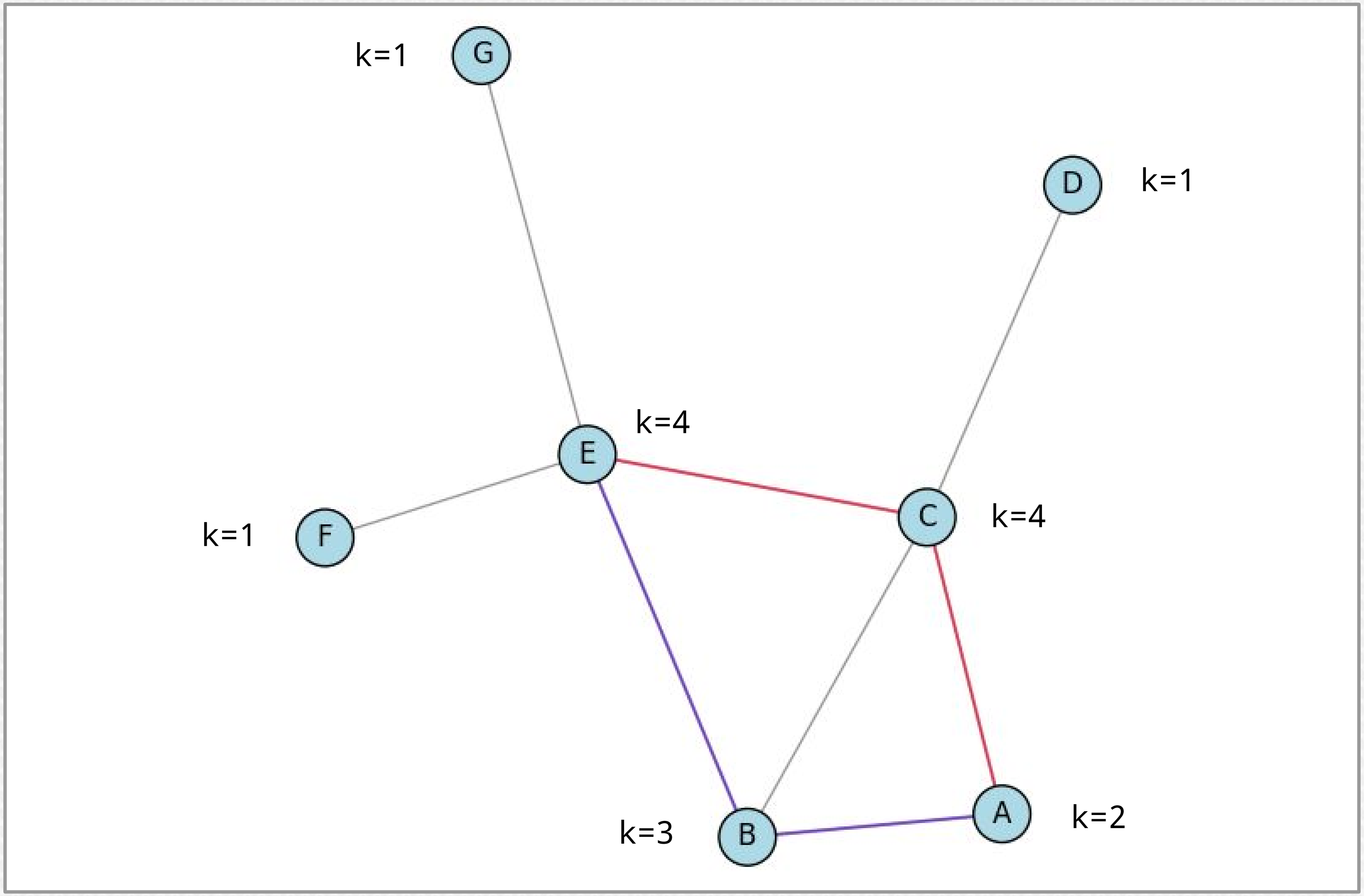
What is the most important vertex?

Centrality

Identification of vertices that play the most important role in a particular network (e.g. how close to the center of the core is the vertex?)

Degree Centrality

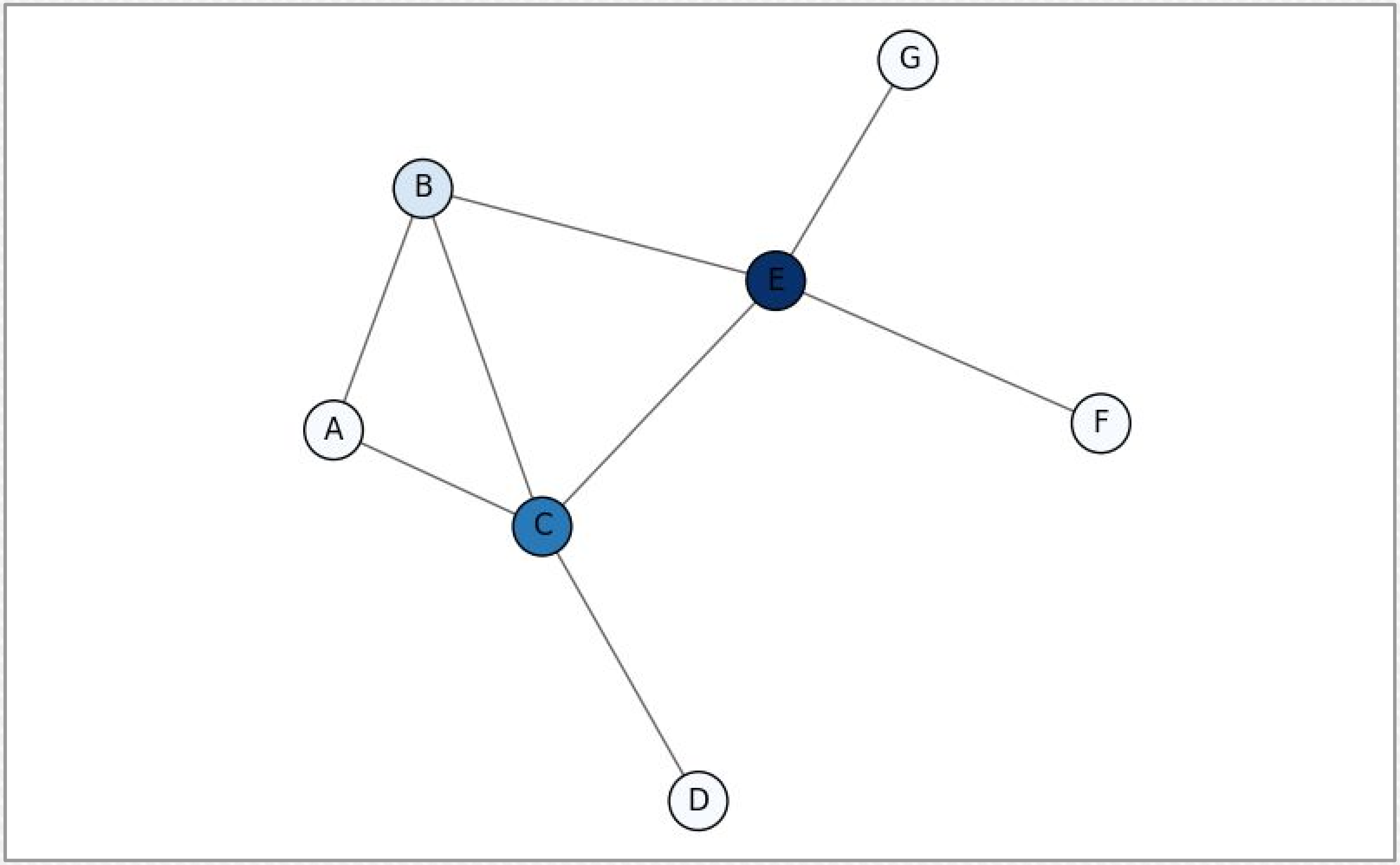
A measure of popularity, determines nodes that can quickly spread information to a localized area.



Degree centrality simply ranks nodes by their degree.

Betweenness Centrality

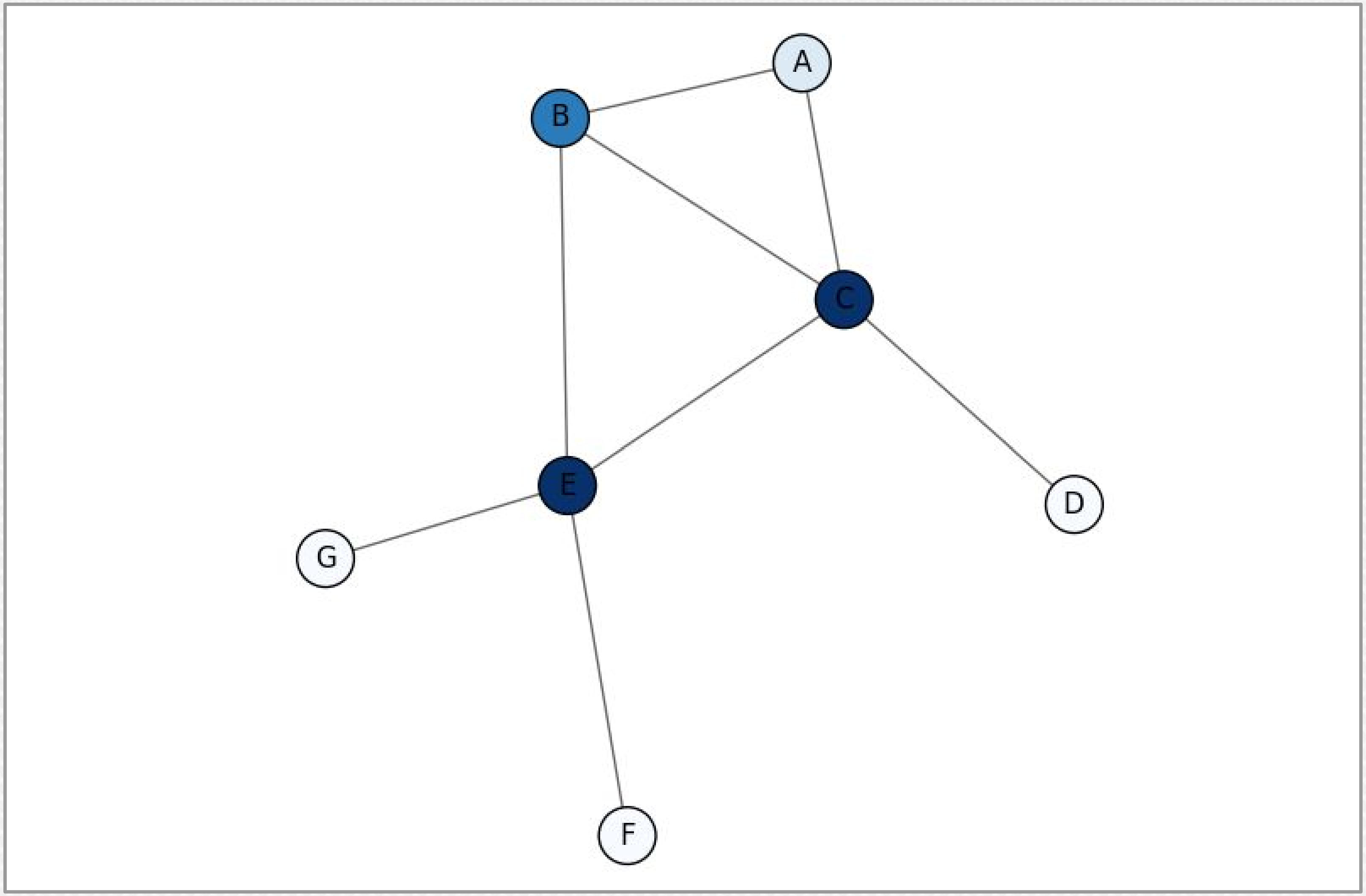
Shows which nodes are likely pathways of information and can be used to determine how a graph will break apart of nodes are removed.



Betweenness: the sum of the fraction of all the pairs of shortest paths that pass through that particular node. Can also be normalized by the number of nodes or an edge weight.

Closeness Centrality

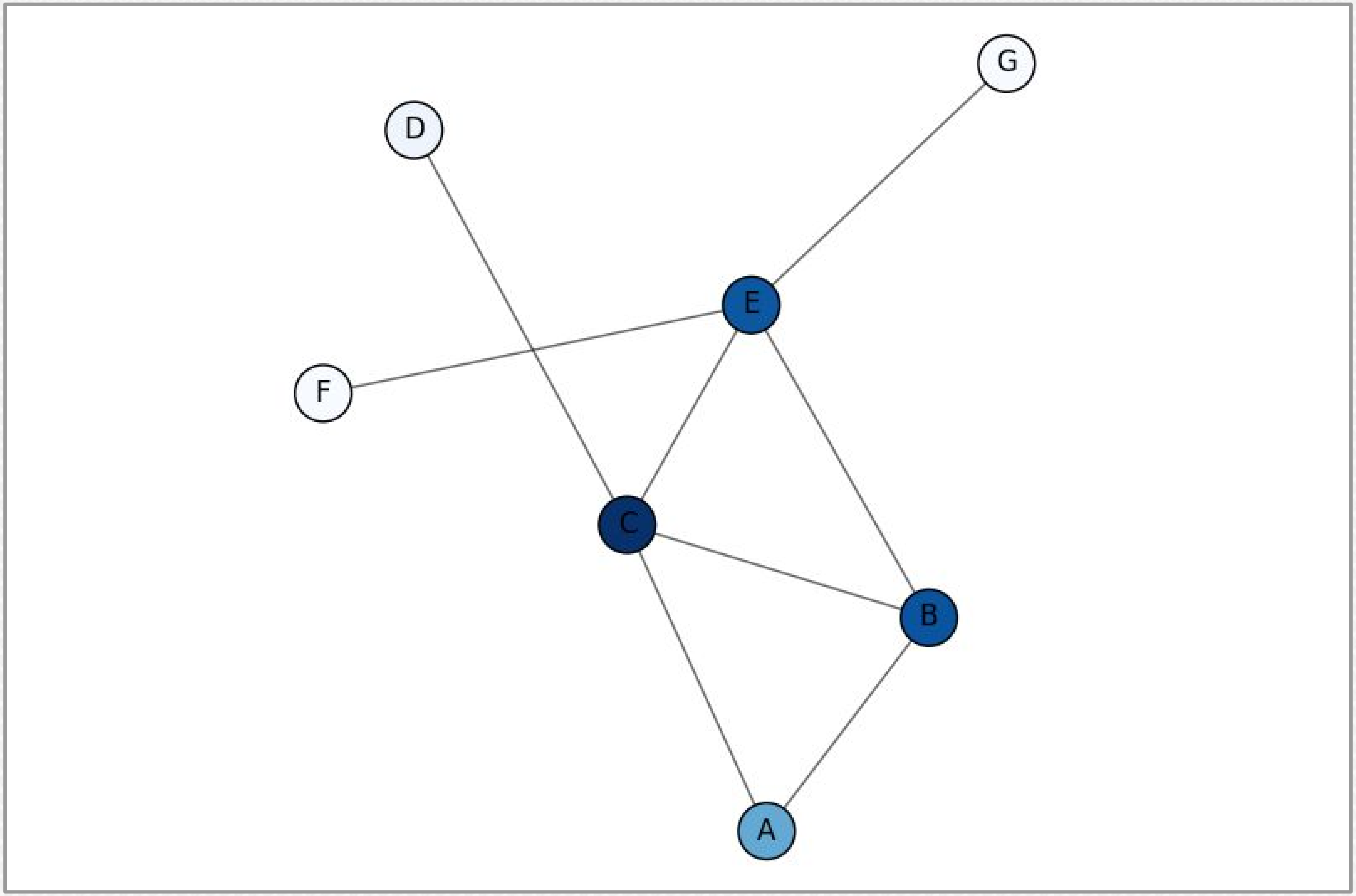
A measure of reach; how fast information will spread to all other nodes from a single node.



Closeness: average number of hops to reach any other node in the network.
The reciprocal of the mean distance: n-1 / size(G) - 1 for a neighborhood, n

Eigenvector Centrality

A measure of related influence, who is closest to the most important people in the Graph? Kind of like “power behind the scenes” or influence beyond popularity.



Eigenvector: proportional to the sum of centrality scores of the neighborhood.
(PageRank is a stochastic eigenvector scoring)

Clustering

Detection of communities or groups that exist in a network by counting triangles.

Measures “transitivity” - tripartite relationships that indicate clusters

`T(i)` = # of triangles with `i` as a vertex

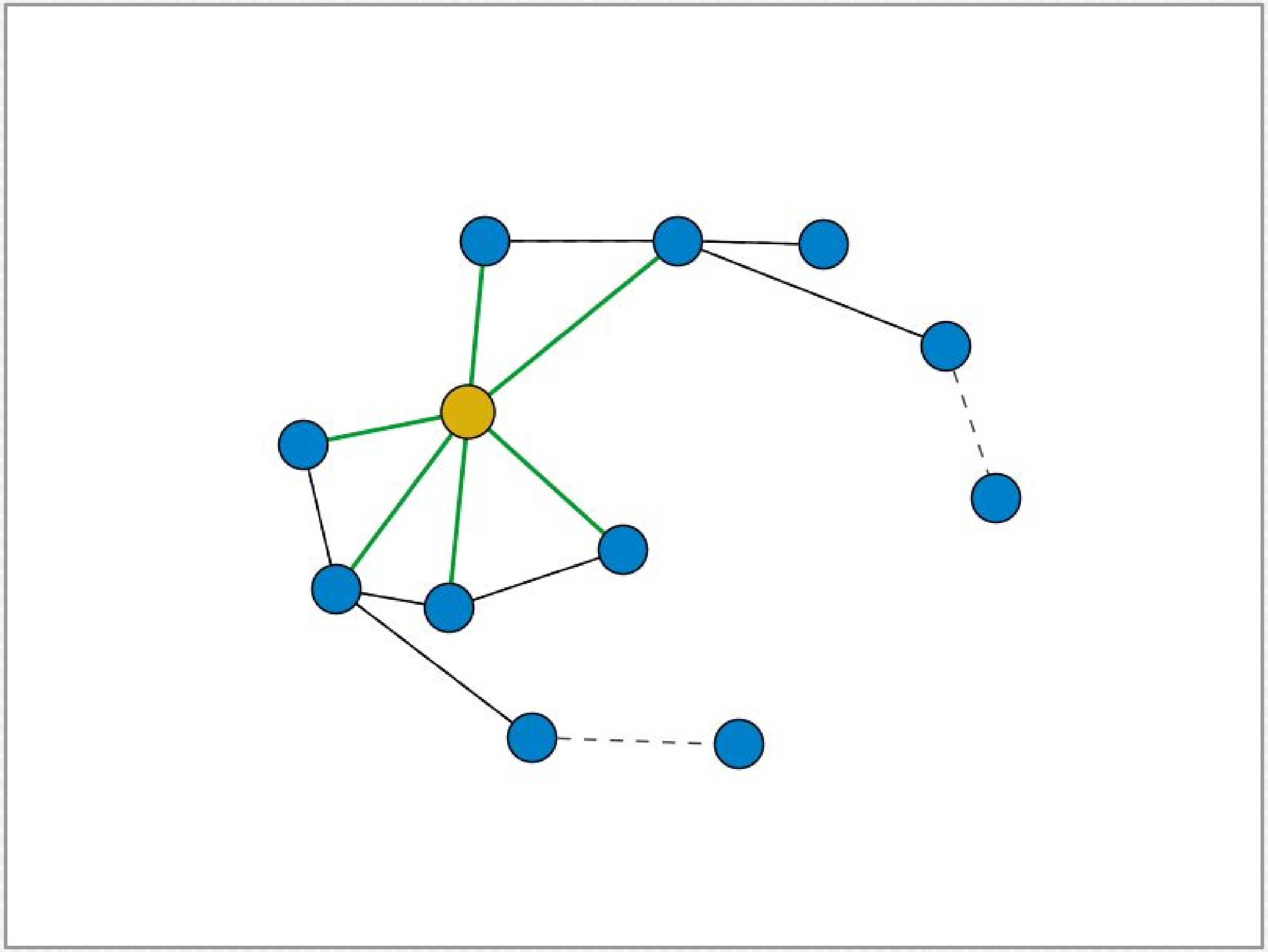
$$C_i = \binom{k_i}{2}^{-1} T(i)$$

Local Clustering Coefficient

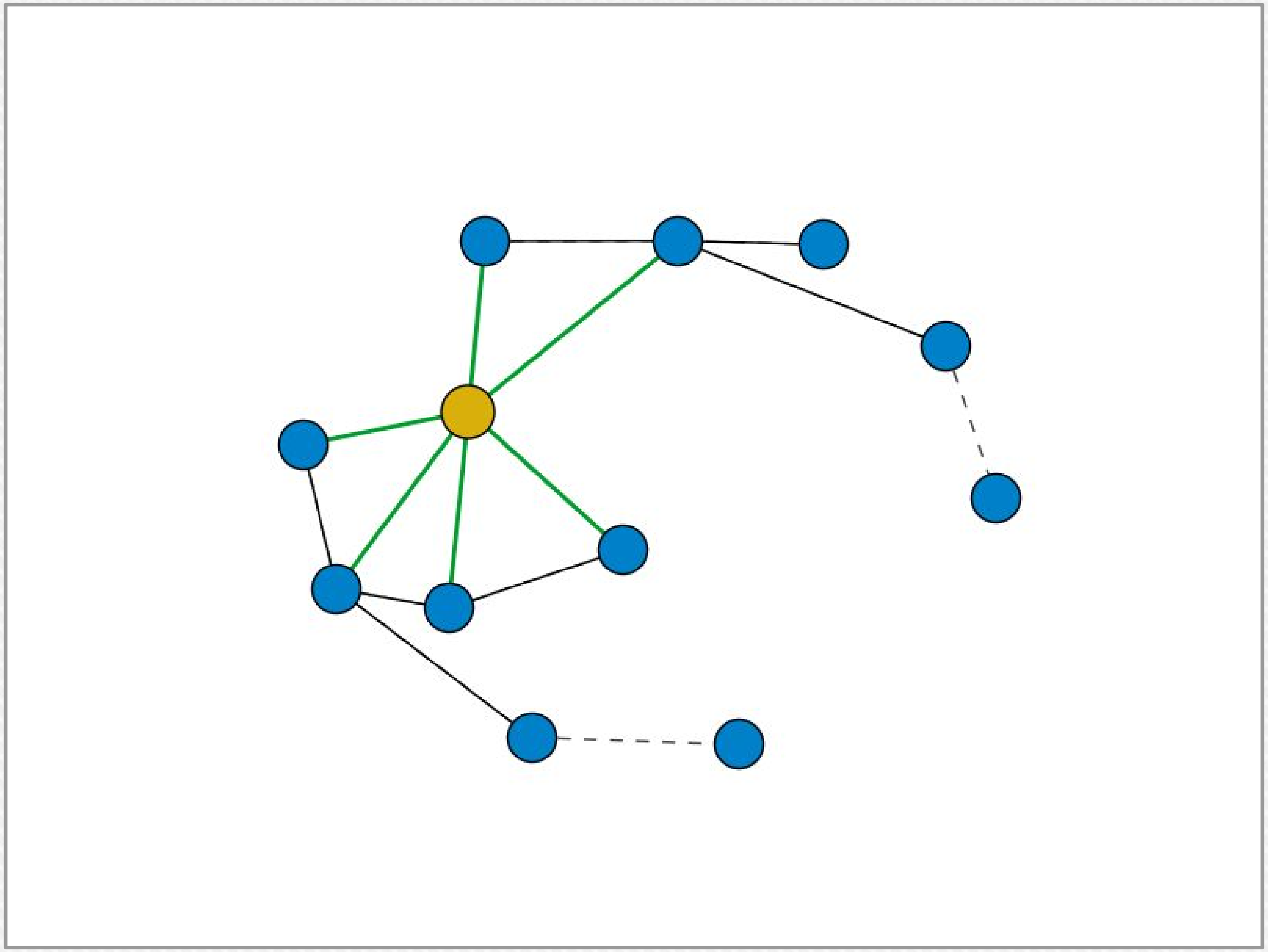
$$C = \frac{1}{n} \sum_{i \in V} C_i$$

Graph Clustering Coefficient

Counting the number of triangles is a start towards making inferences about “transitive closures” - e.g. predictions or inferences about relationships



Green lines are connections from the node, black are the other connections

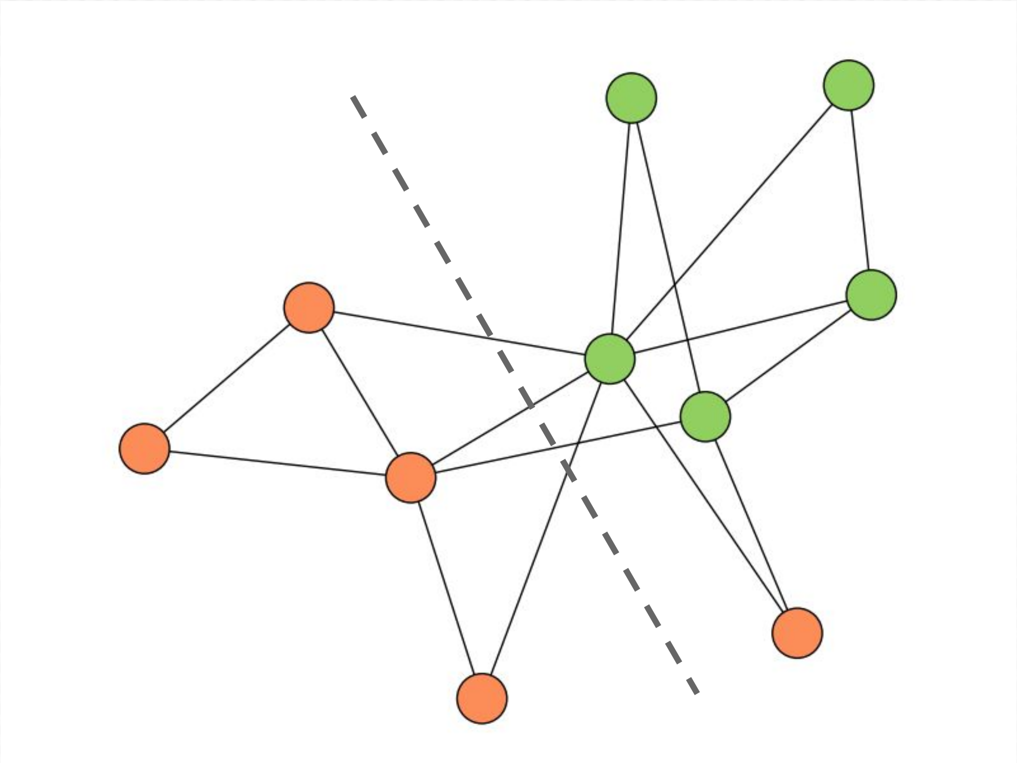


$k_i = 6$

$T(i) = 4$

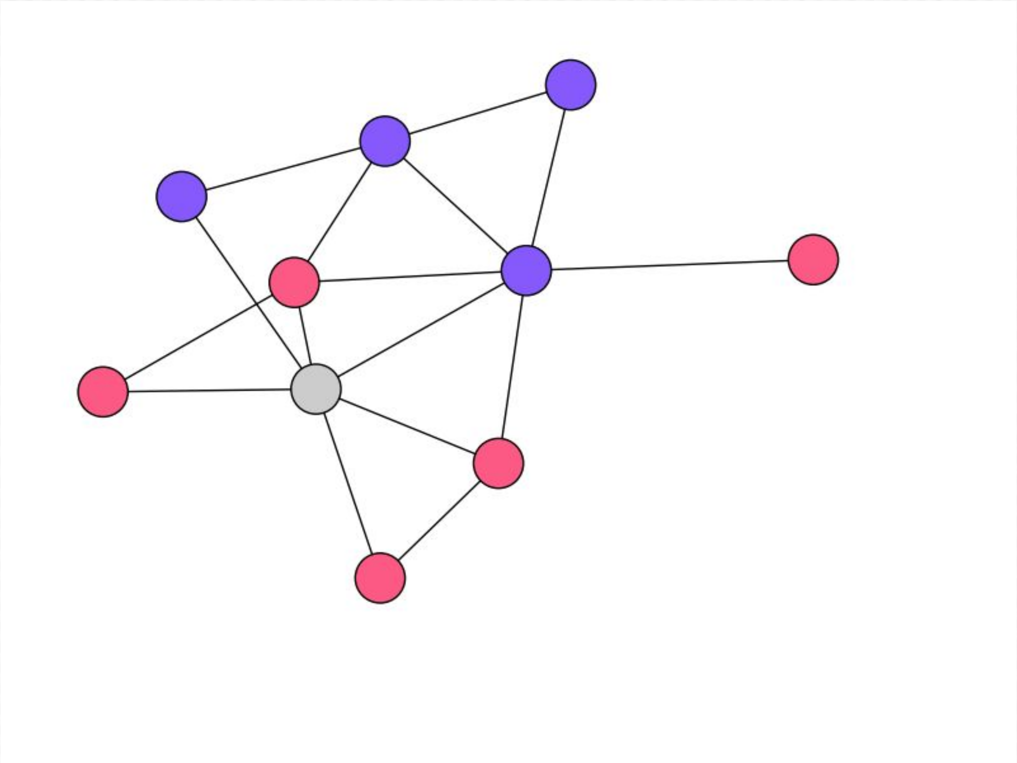
$C_i = (2*4) / (6*(6-1)) = 0.266$

Classification



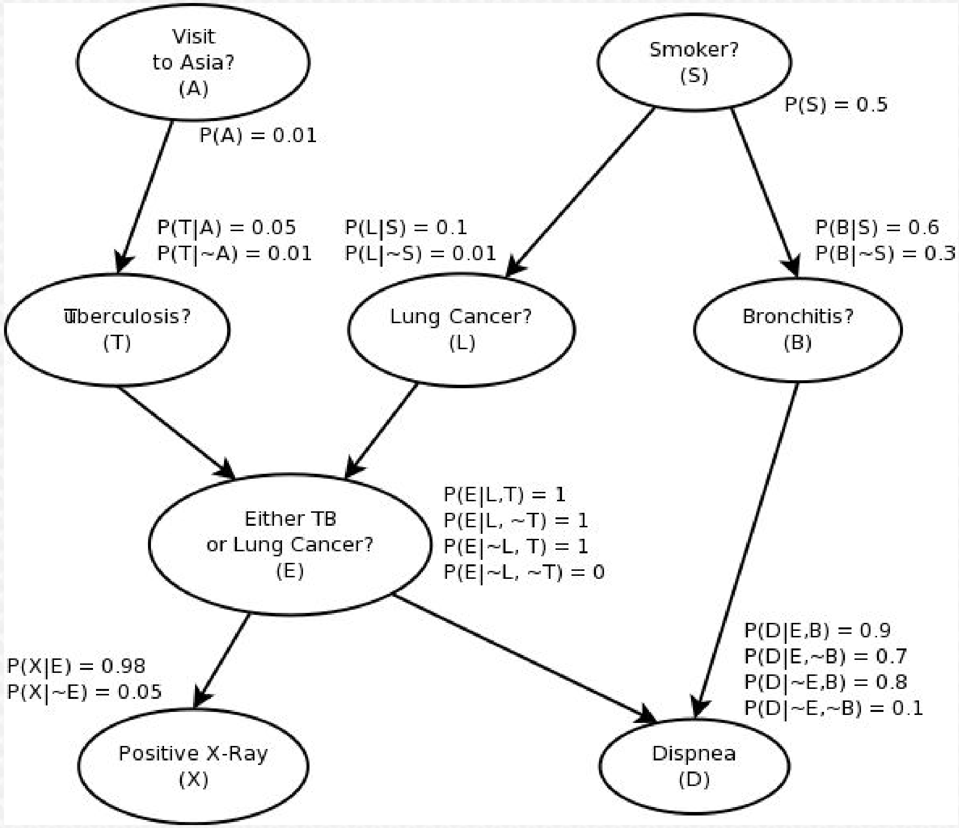
Partitive classification utilizes subgraphing techniques to find the minimum number of splits required to divide a graph into two classes. Laplacian Matrices are often used to count the number of spanning trees.

Classification



Distance based techniques like k-Nearest Neighbors embed distances in graphical links, allowing for very fast computation and blocking of pairwise distance computations.

Bayesian Networks



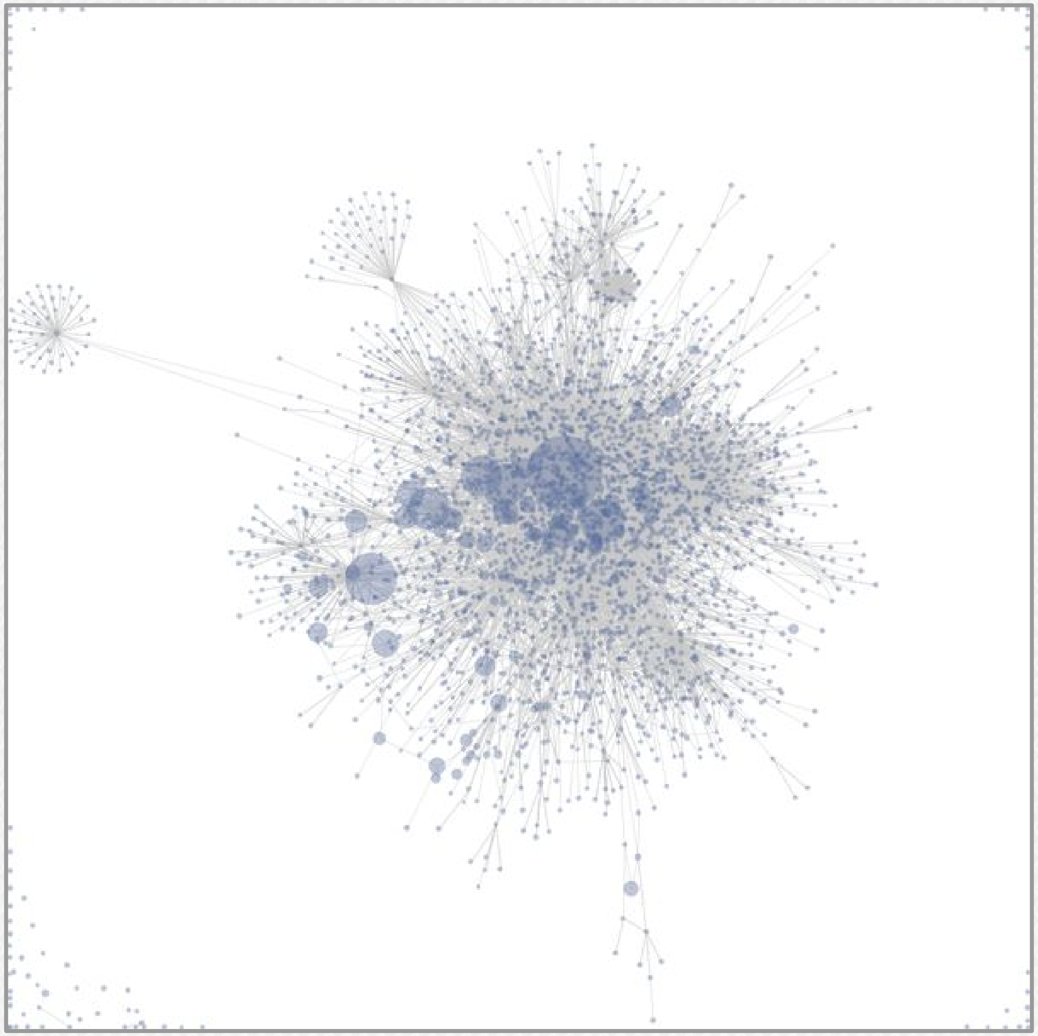
Just add probability! Bayesian Networks are directed, acyclic graphs that encode conditional dependencies and can be trained from data, then used to make inferences.

Network Visualization

Layouts

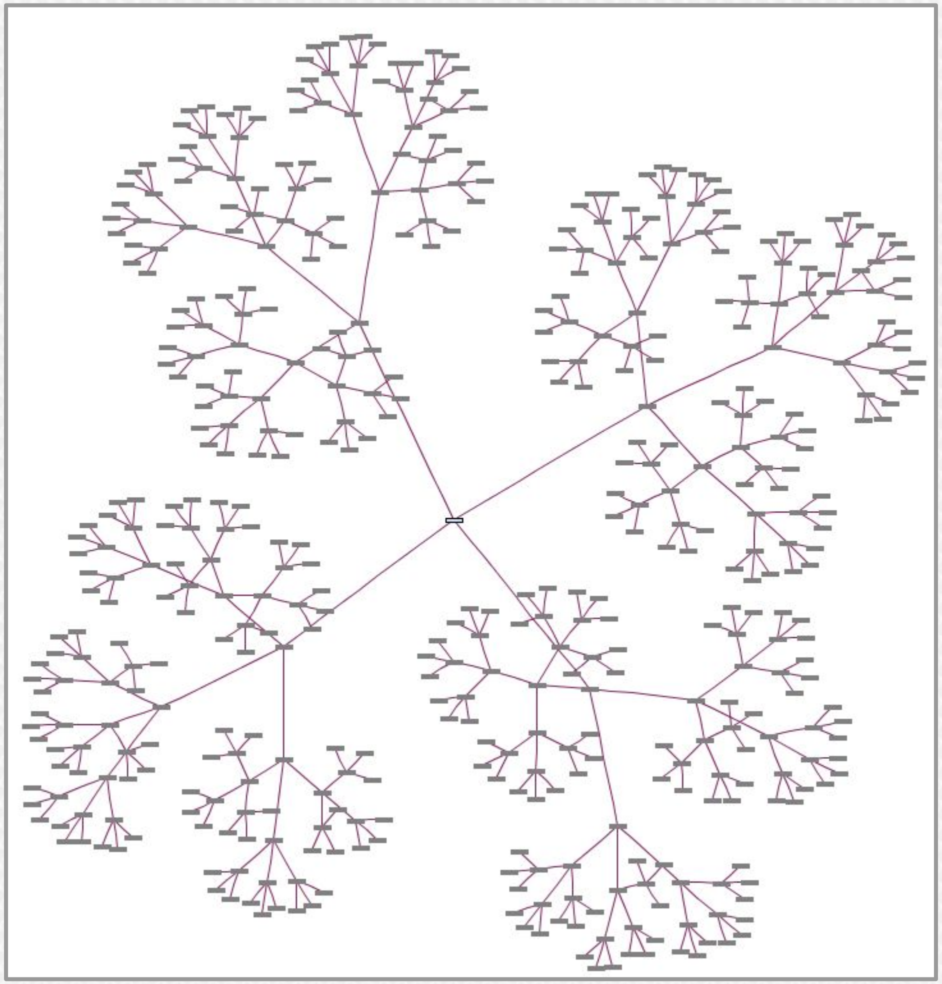
- Open Ord
  - http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=731088
  - Draws large scale undirected graphs with visual clusters
- Yifan Hu
  - http://yifanhu.net/PUB/graph_draw_small.pdf
  - Force Directed Layout with multiple levels and quadtree
- Force Atlas
  - http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0098679
  - A continuous force directed layout (default of Gephi)
- Fruchterman Reingold
  - http://cs.brown.edu/~rt/gdhandbook/chapters/force-directed.pdf
  - Graph as a system of mass particles (nodes are particles, edges are springs) This is the basis for force directed layouts

Others: circular, shell, neato, spectral, dot, twopi …



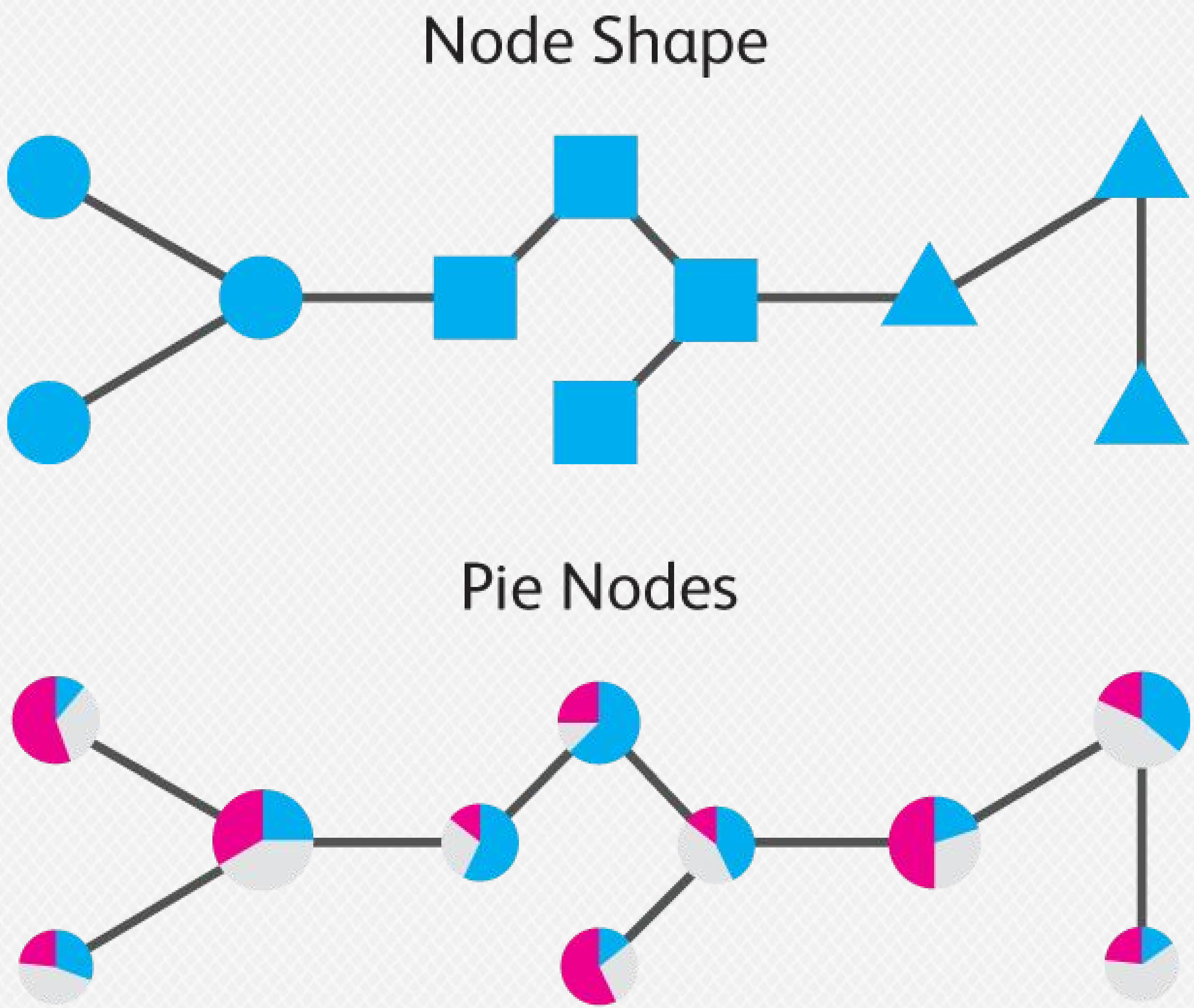
Force Directed

http://en.wikipedia.org/wiki/Force-directed_graph_drawing



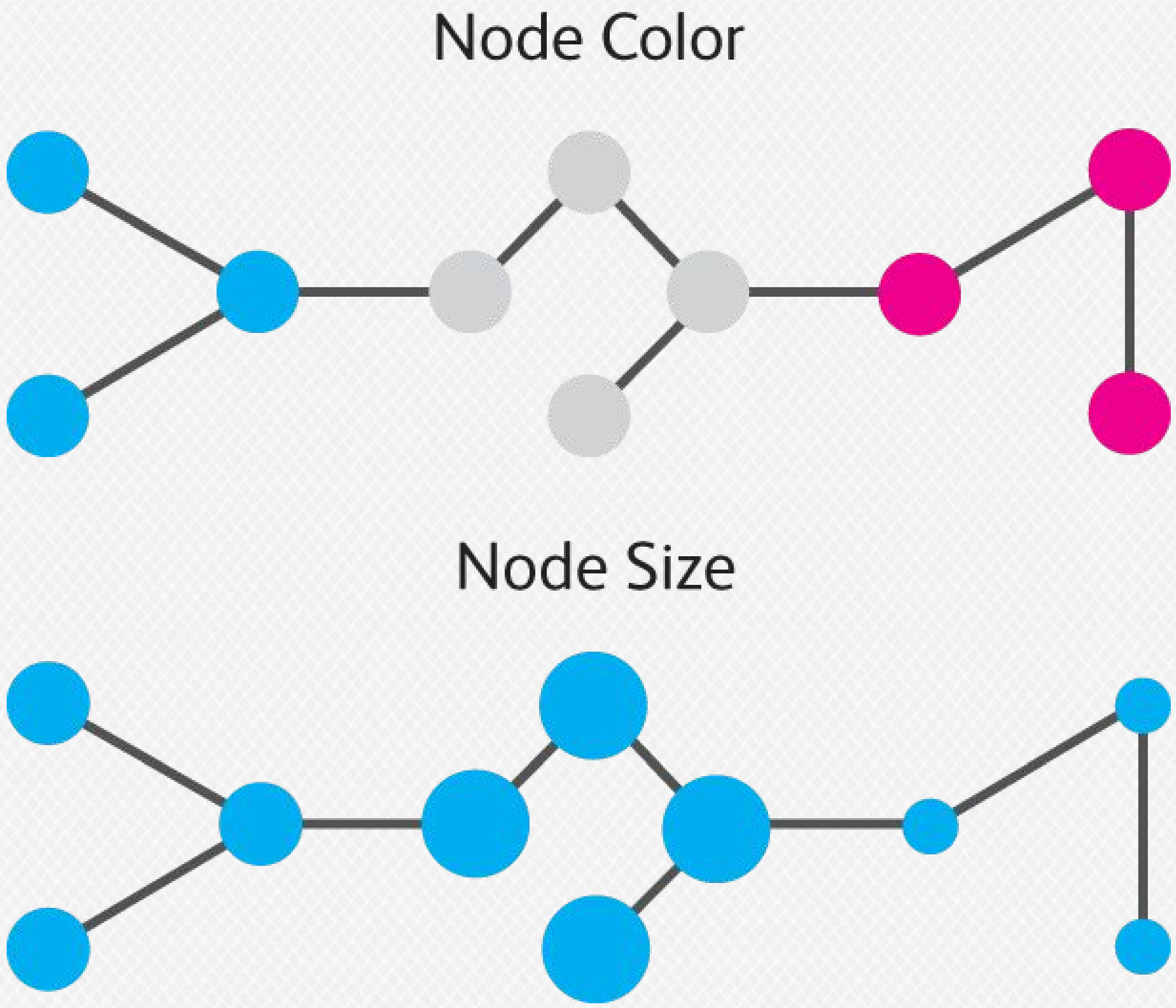
Hierarchical Graph Layout

https://seeingcomplexity.files.wordpress.com/2011/02/tree_graph_example.gif



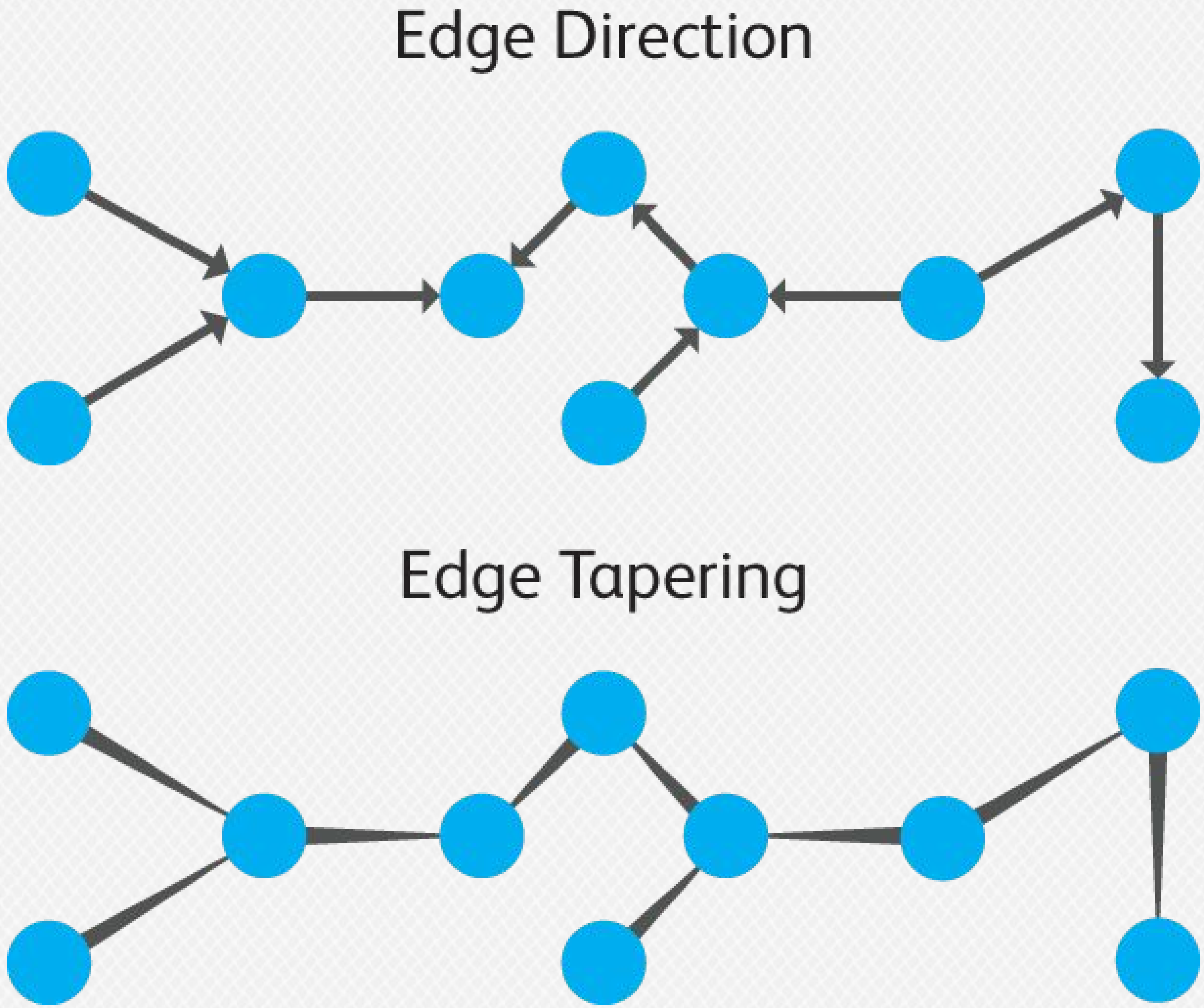
Lane Harrison, The Links that Bind Us: Network Visualizations

http://blog.visual.ly/network-visualizations



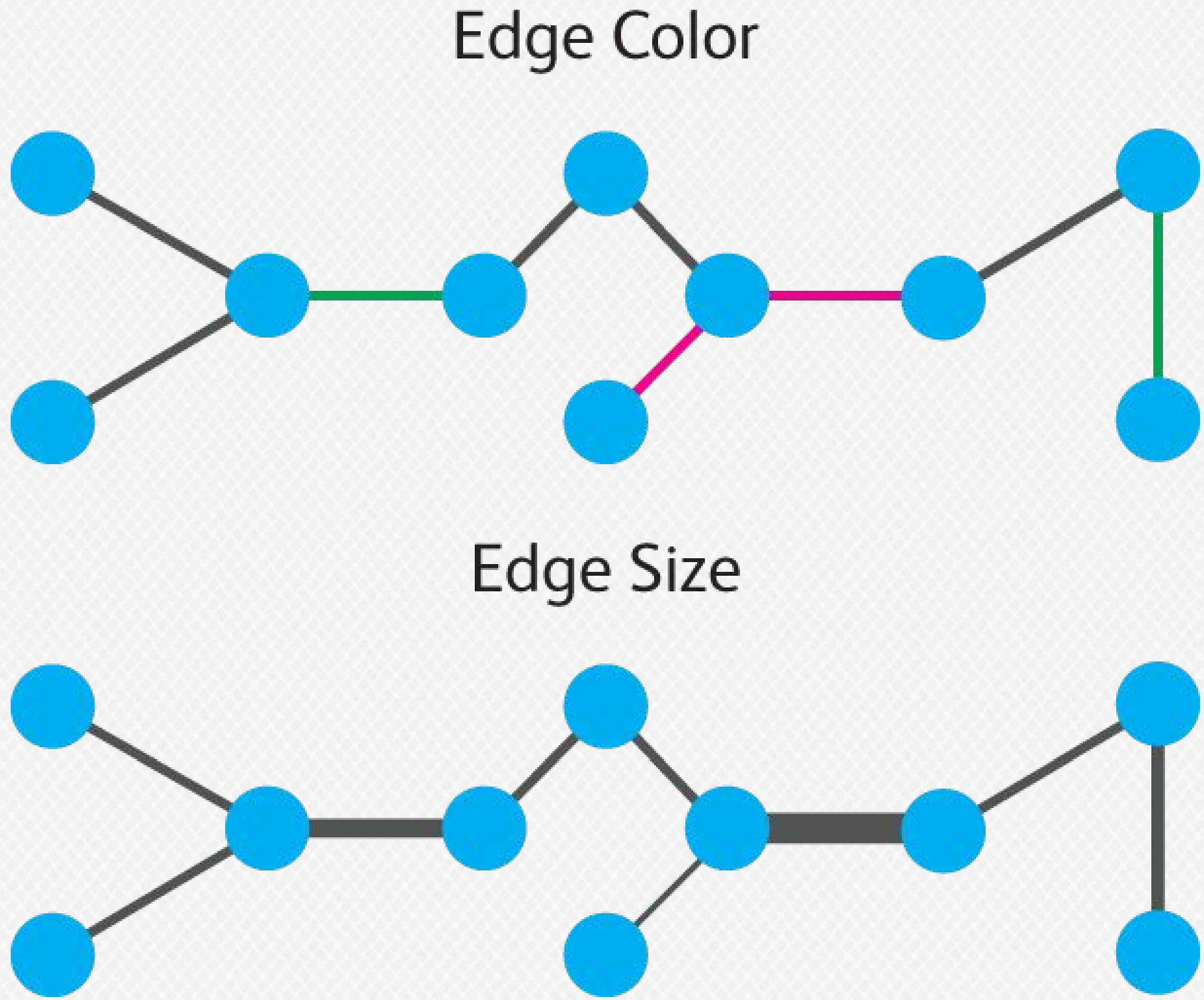
Lane Harrison, The Links that Bind Us: Network Visualizations

http://blog.visual.ly/network-visualizations



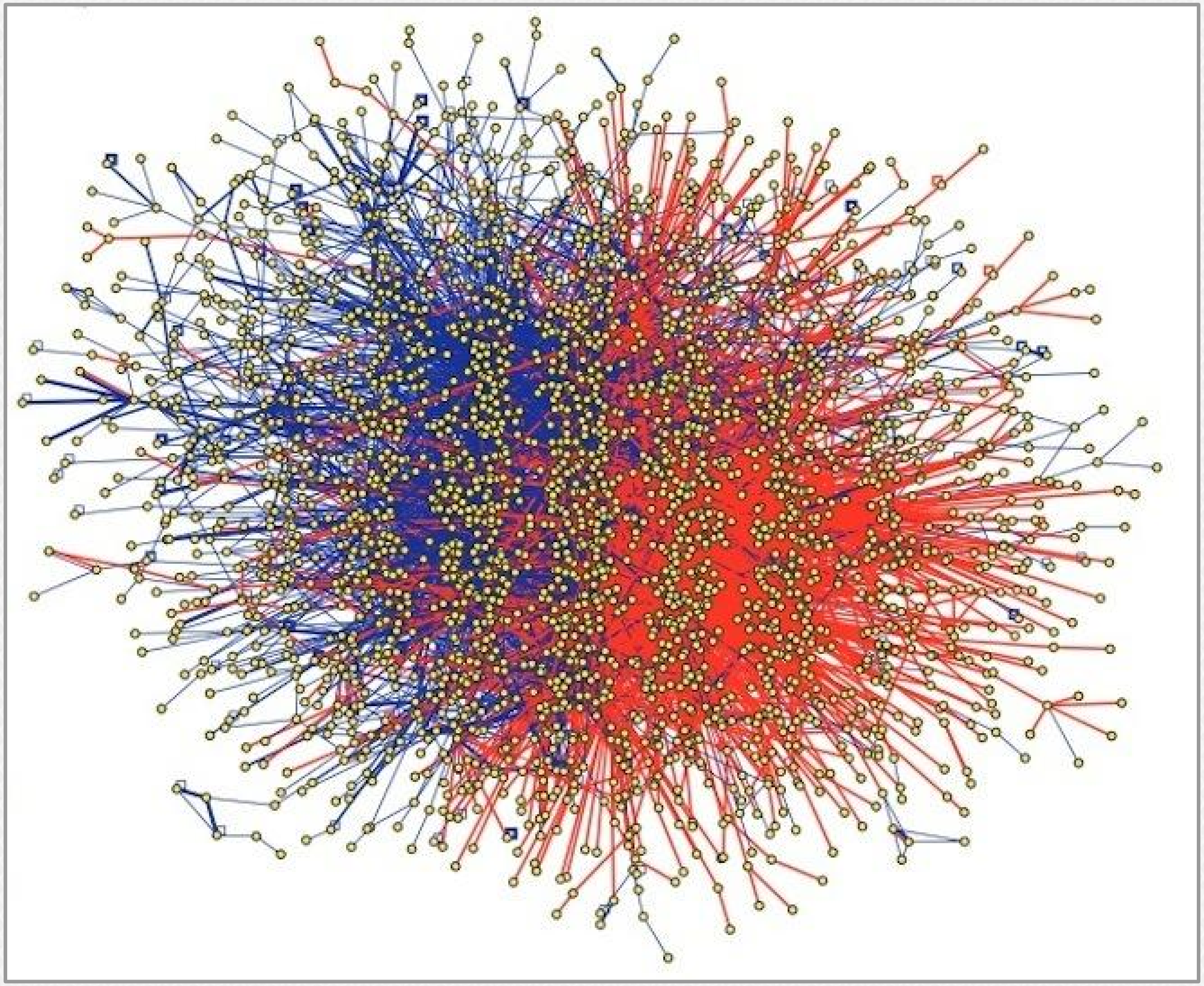
Lane Harrison, The Links that Bind Us: Network Visualizations

http://blog.visual.ly/network-visualizations



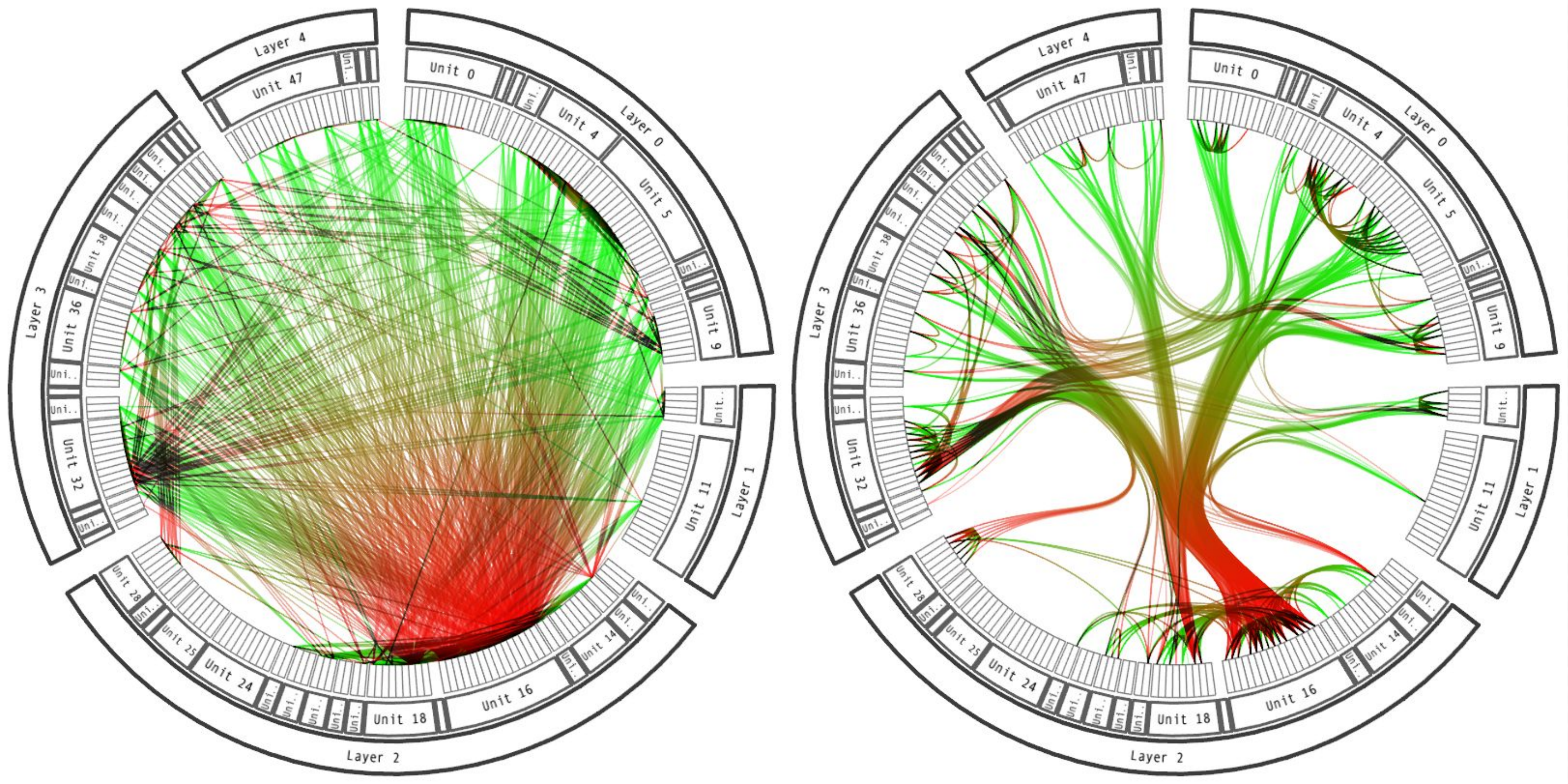
Lane Harrison, The Links that Bind Us: Network Visualizations

http://blog.visual.ly/network-visualizations



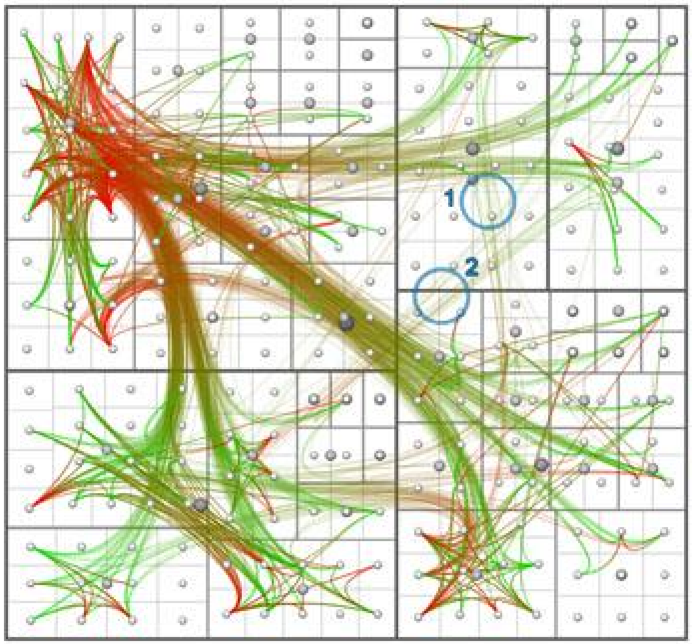
The Hairball

http://www.slideshare.net/OReillyStrata/visualizing-networks-beyond-the-hairball



Edge Bundling

https://seeingcomplexity.wordpress.com/2011/02/05/hierarchical-edge-bundles/



Region Bundling

http://infosthetics.com/archives/2007/03/hierarchical_edge_bundles.html

Tools for Graph Visualization

Plan of Study

Extraction of Network from Email
Introduction to NetworkX
Analyzing our Email Networks
Visualizing our Email Network
Relief from Gephi

tribe

~/Repos/ddl/tribe — -bash

```
usage: tribe-admin.py [-h] [-v] {headers,count,extract,info,draw} ...

An administrative utility for the Tribe Social Network Analysis

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit

commands:
  Administrative commands for Tribe

  {headers,count,extract,info,draw}
    headers             Perform an analysis of the email headers in an MBox
    count               Count the number of emails in an MBox.
    extract             Extract a GraphML file from an MBox
    info                Print information about a GraphML file
    draw                Draw a GraphML using the tribe draw method

If there are any bugs or concerns, submit an issue on Github
(tribe) apollo:tribe benjamin$ ./tribe-admin.py extract --help
usage: tribe-admin.py extract [-h] [-w WRITE] mbox

positional arguments:
  mbox                  Path or location to MBox for analysis

optional arguments:
  -h, --help            show this help message and exit
  -w WRITE, --write WRITE
                        Location to write data to
(tribe) apollo:tribe benjamin$ ./tribe-admin.py extract fixtures/benjamin\@bengfort.com.mbox -w bbengfort.graphml
Starting Graph extraction, could take time ...
Graph extraction took 445.658 seconds
```

Graph Extraction from an email MBox