Graph Networks

Putting the Relationships back in Databases

Name
Date

// FLATIRON SCHOOL

What are Graphs?

Concepts

- Terminology
- Types

Case Studies

Graph Algorithms

- Pathfinding
- Centrality Measures
- Clustering

Practical Example

Social Networks

Tweet Data

```
{       Tweet_Id : int,
        Text: str,
        Mentions: str,
        Retweet_id: int,
        Media: str,
        User_id: int
}
```

User Data

```
{       User_id: int,
        Username: str,
        Followers: [‘str’]
}
```

Social Data

My friends are your friends are friends of your friends

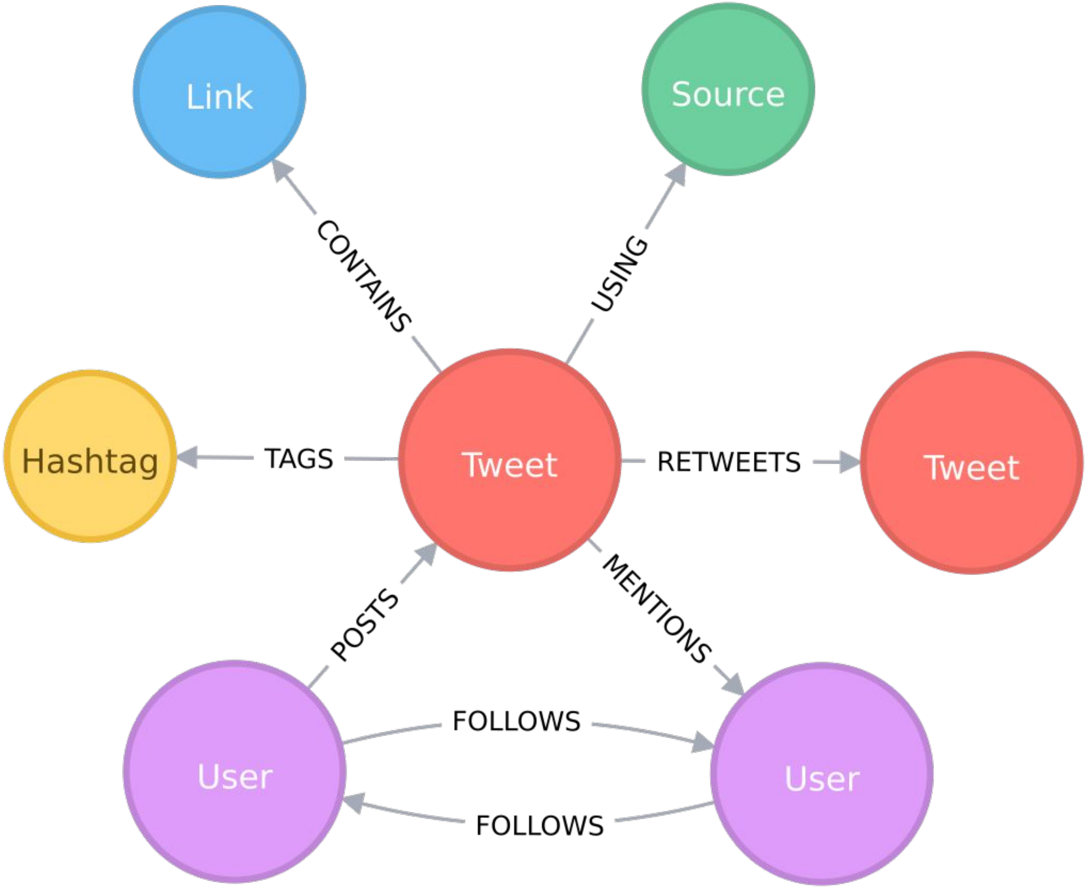
Are there any foreseeable problems with how the data is structured?

Likely questions for the data:

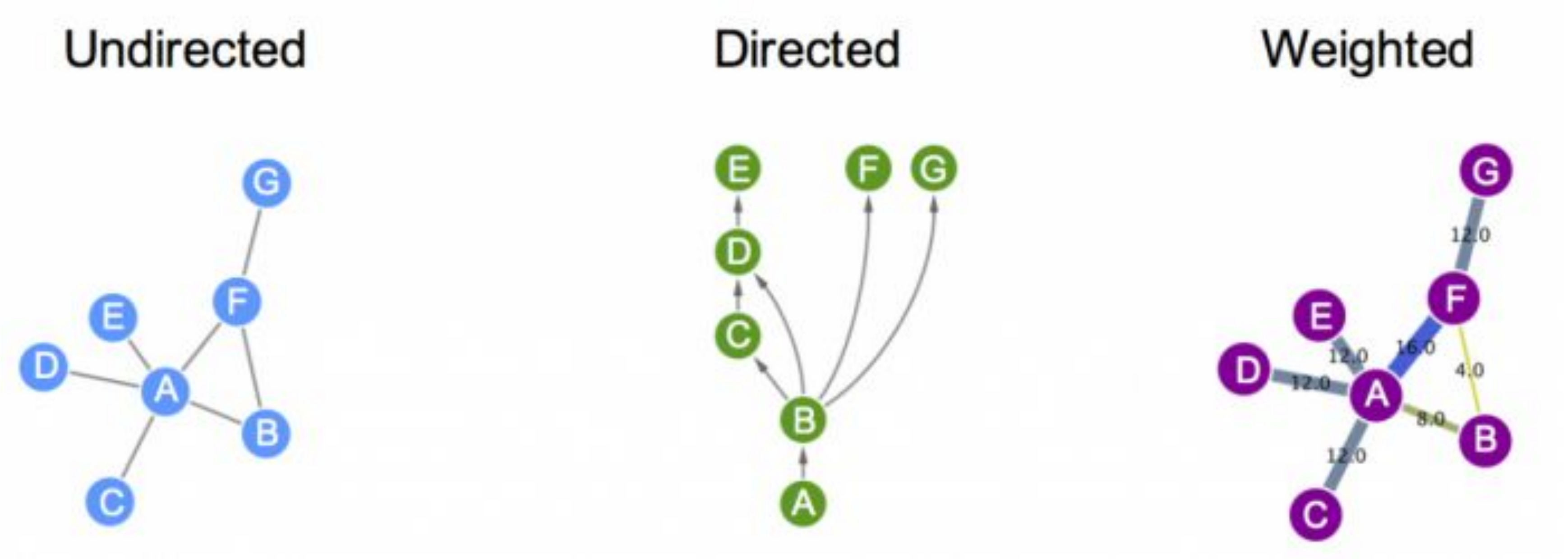
- **How many degrees of separation are there between two users?**
- **Who is the most influential user in the network?**
- **Are there cliques that have formed in the community?**

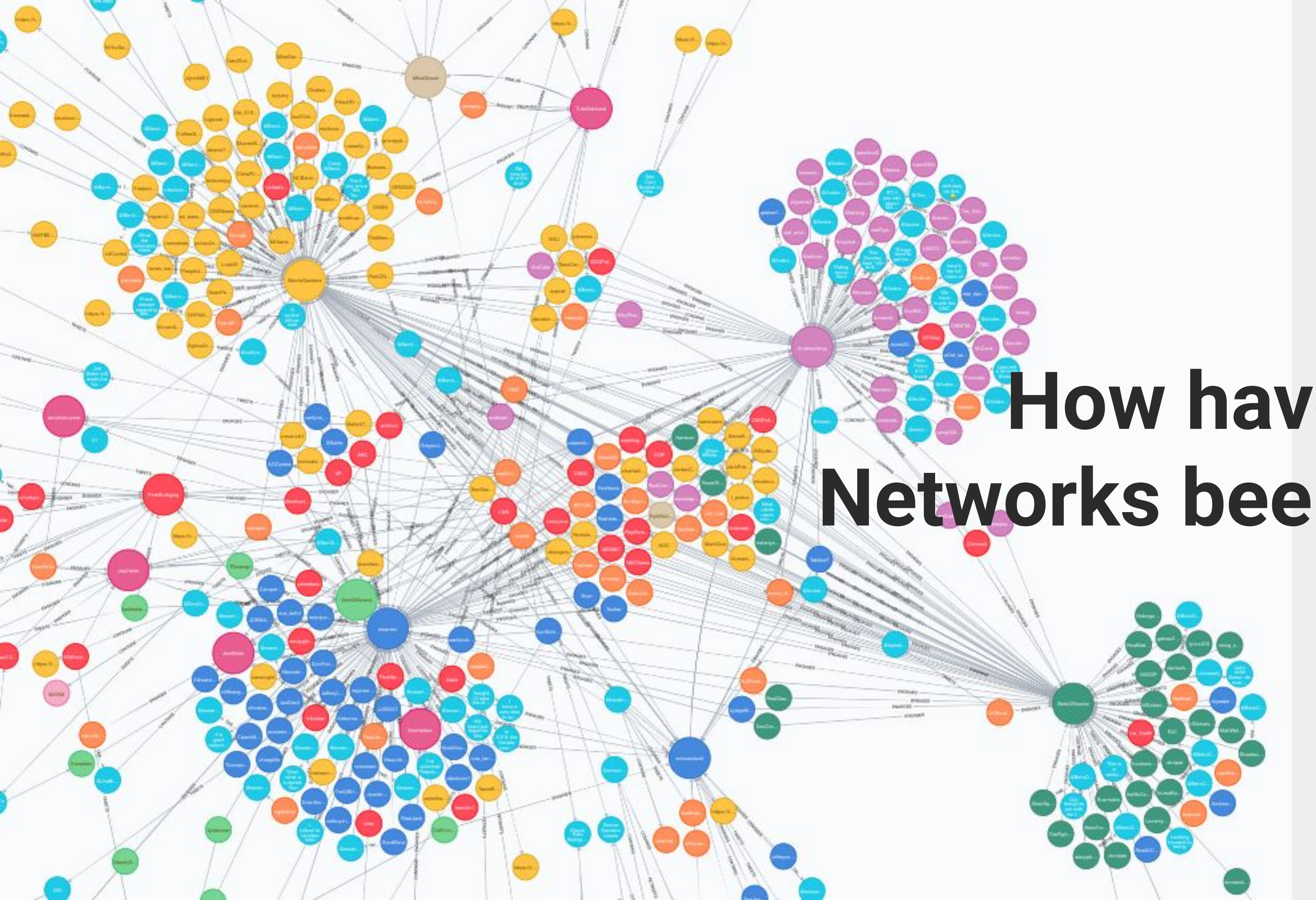
Terminology

- Nodes - Nouns
- Edges - Verbs
- Degree - Connections



Types


Undirected
Directed
Weighted
G
F
E
D
A
B
C
E
F
G
D
C
B
A
G
12.0
F
E
12.0
16.0
D
12.0
4.0
A
8.0
B
12.0
C

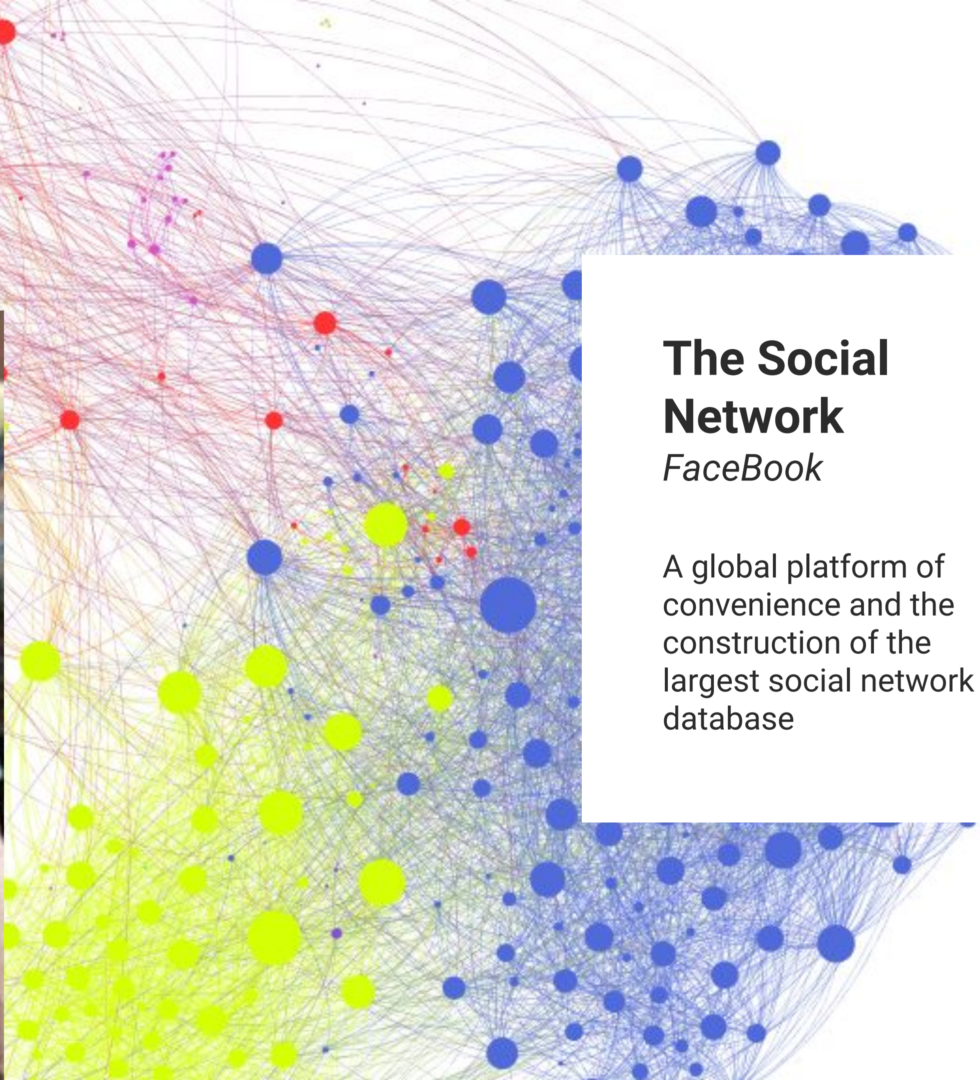


How have Graph Networks been used?

Page Rank

How Google made search better

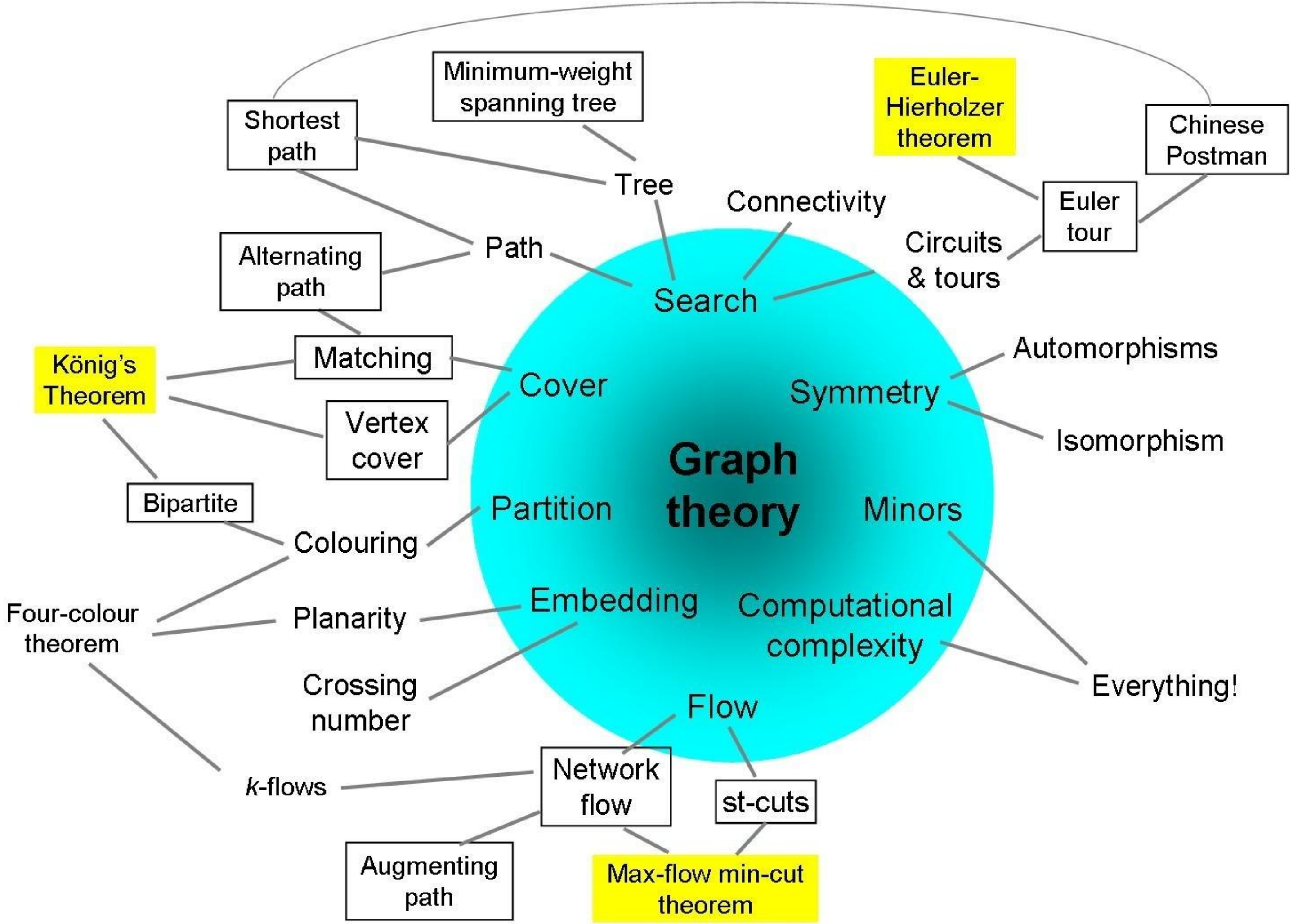
A novel graph algorithm contributed to the birth of a multi-billion dollar enterprise



The Social Network

FaceBook

A global platform of convenience and the construction of the largest social network database



Minimum-weight spanning tree

Shortest path

Tree

Connectivity

Euler-Hierholzer theorem

Chinese Postman

Euler tour

Circuits & tours

Path

Alternating path

Search

Matching

König's Theorem

Cover

Automorphisms

Symmetry

Vertex cover

Isomorphism

Graph theory

Bipartite

Partition

Minors

Colouring

Embedding

Computational complexity

Four-colour theorem

Planarity

Crossing number

Everything!

Flow

Network flow

st-cuts

k-flows

Augmenting path

Max-flow min-cut theorem

Pathfinding - How can you get from one node to another?

Applications

- Distance from node to node
- Analyzing routes from node to node

Metrics

- Degree separation
  - Bacon
- Euclidean
  - Weights

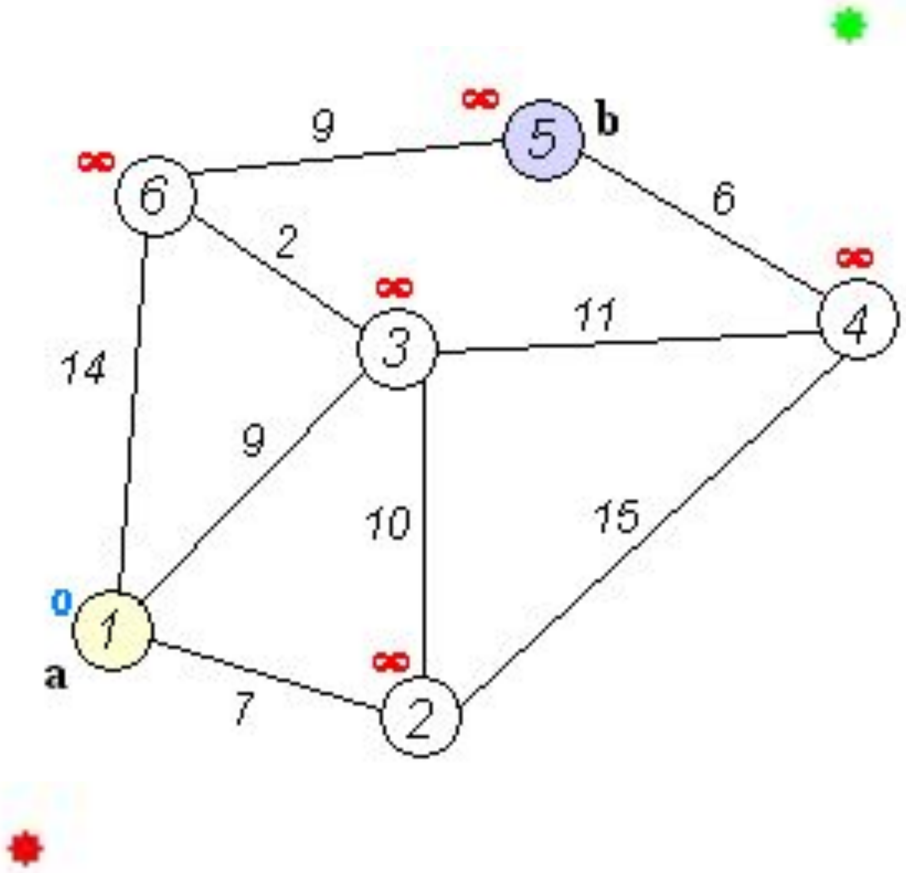
Algorithms

- A*
- Random Walk
- Shortest Path
  - Djiskstra

Shortest Path First

> “ What is the shortest way to travel from Rotterdam to Groningen, in general: from given city to given city. It is the algorithm for the shortest path, which I designed in about twenty minutes. One morning I was shopping in Amsterdam with my young fiancée, and tired, we sat down on the café terrace to drink a cup of coffee and I was just thinking about whether I could do this, and I then designed the algorithm for the shortest path . . . One of the reasons that it is so nice was that I designed it without pencil and paper. I learned later that one of the advantages of designing without pencil and paper is that you are almost forced to avoid all avoidable complexities. ”

— Edsger Dijkstra, interview



Centrality - What are the influential nodes in the network?

Applications

- Determine prominence of node
- Figure out sparseness of data

Metrics

- Degree
  - Normalized connections
- Betweenness

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

- Closeness

$$C(x) = \frac{1}{\sum_y d(y,x)}.$$

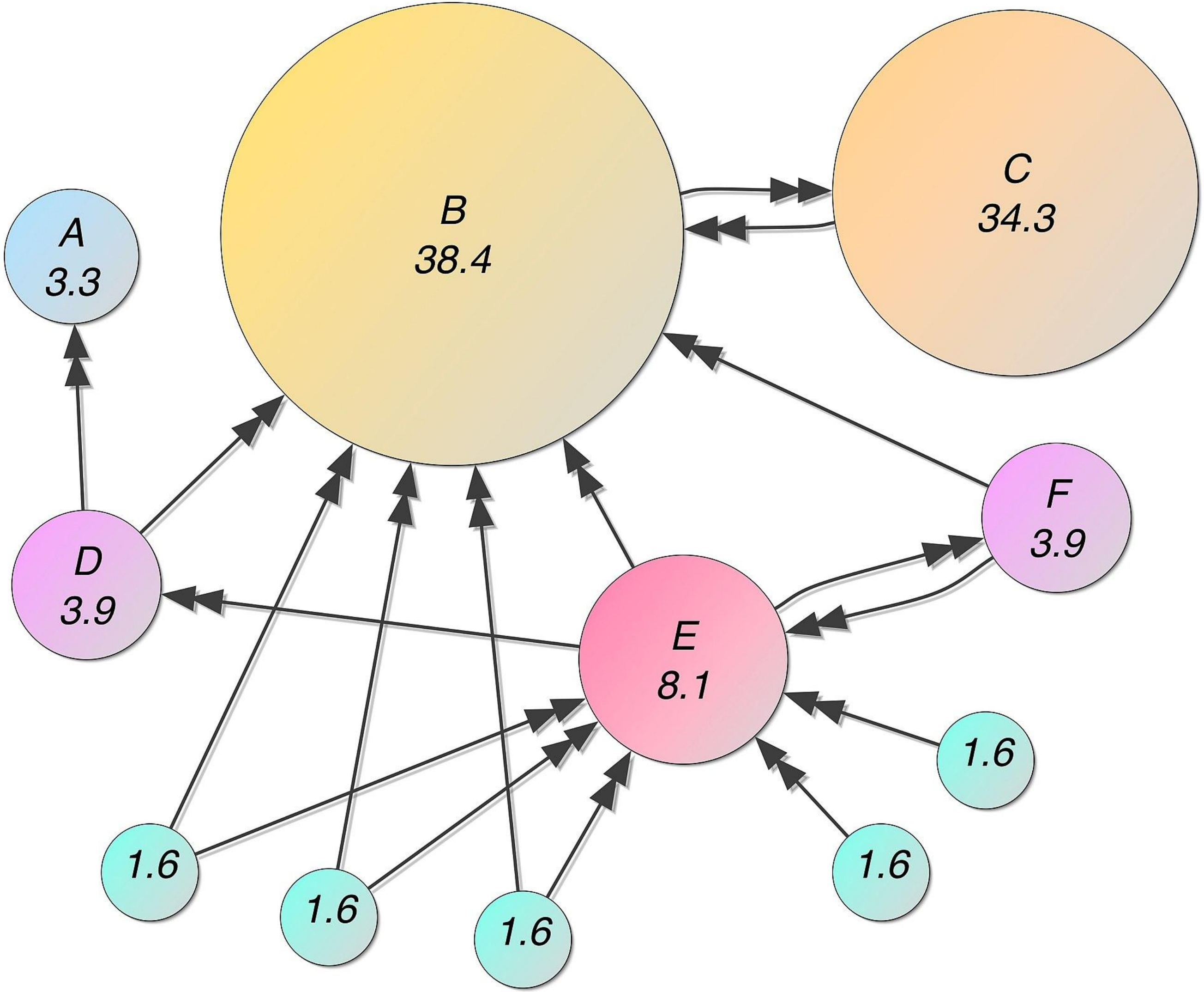
Algorithms

- Page Rank
  - Deep dive

PageRank

PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

— Some Person, Wikipedia



PageRank

```
# Parameter M adjacency matrix where M_i,j represents the link from 'j'
to 'i', such that for all 'j'
# sum(i, M_i,j) = 1
# Parameter d damping factor (default value 0.85)
# Parameter eps quadratic error for v (default value 1.0e-8)
# Return v, a vector of ranks such that v_i is the i-th rank from [0, 1]

import numpy as np

def pagerank(M, eps=1.0e-8, d=0.85):
    N = M.shape[1]
    v = np.random.rand(N, 1)
    v = v / np.linalg.norm(v, 1)
    last_v = np.ones((N, 1), dtype=np.float32) * 100

    while np.linalg.norm(v - last_v, 2) > eps:
        last_v = v
        v = d * np.matmul(M, v) + (1 - d) / N
    return v

M = np.array([[0, 0, 0, 0, 1],
              [0.5, 0, 0, 0, 0],
              [0.5, 0, 0, 0, 0],
              [0, 1, 0.5, 0, 0],
              [0, 0, 0.5, 1, 0]])
v = pagerank(M, 0.001, 0.85)
```

$$PageRank\ of\ site = \sum \frac{PageRank\ of\ inbound\ link}{Number\ of\ links\ on\ that\ page}$$

OR

$$PR(u) = (1 - d) + d \times \sum \frac{PR(v)}{N(v)}$$

Clustering - Are there distinct groups within the network?

Applications

- Community Detection
- Recommendations

Metrics

- Clustering coefficient
- Cliques

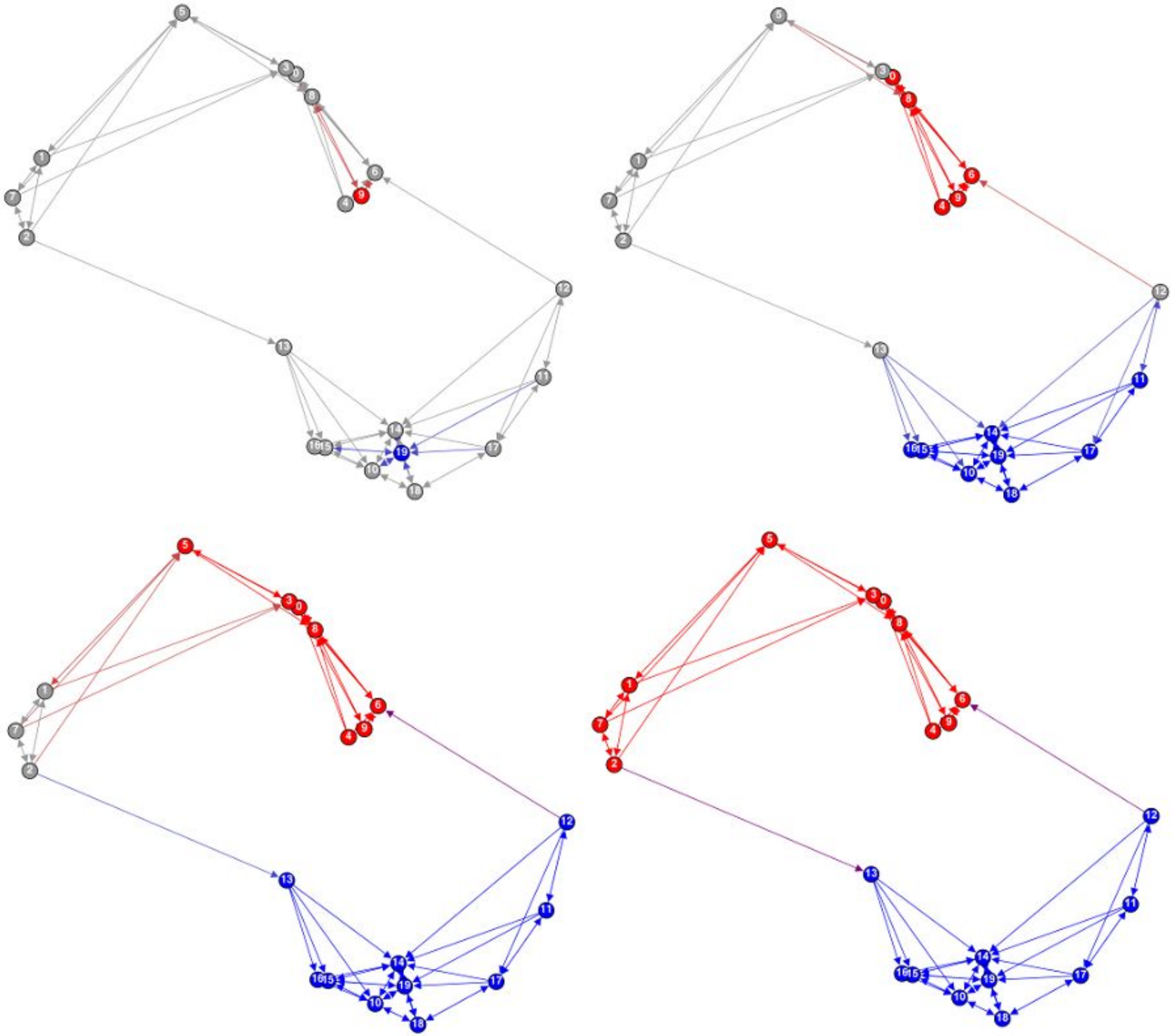
Algorithms

- Label Propagation

Label Propagation

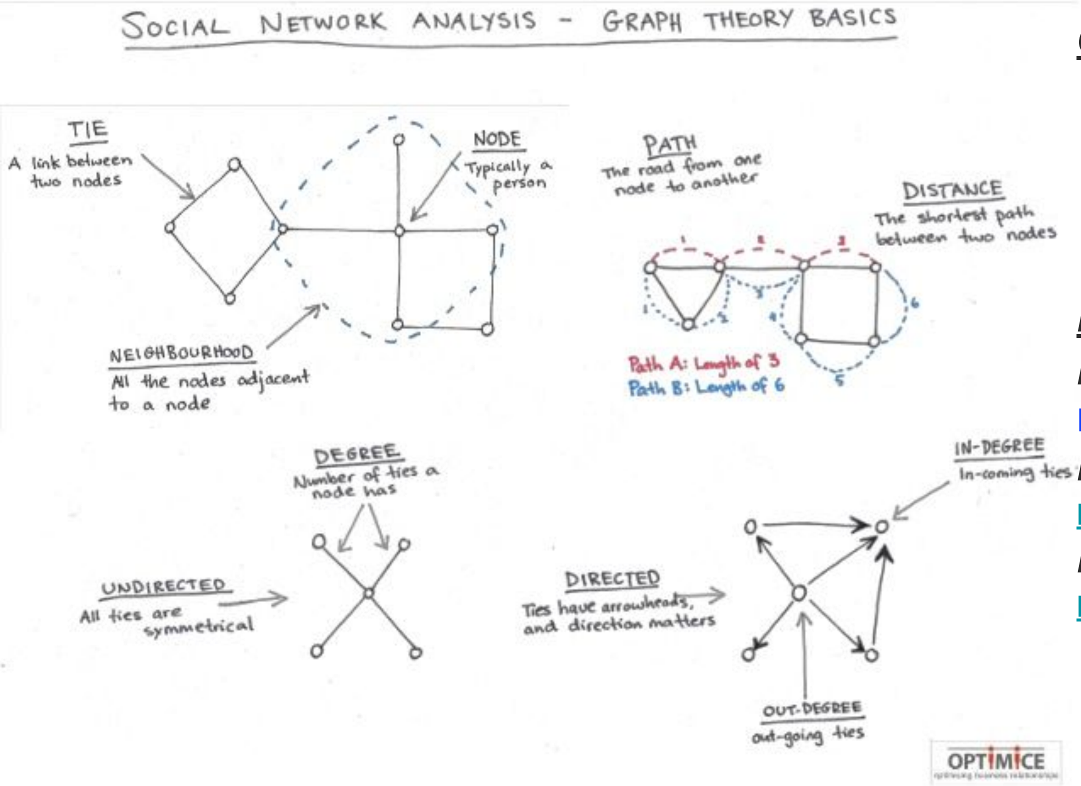
Think K-nearest neighbors, but with actual neighbors

https://freecontent.manning.com/poor-mans-training-data-graph-based-semi-supervised-learning/



Graph Networks

Review



Graph Theory

- **What is a graph network?**
- **What metrics are used/created for networks?**
- **What aspects of data can graphs capture?**

Resources

Project Ideas

http://snap.stanford.edu/class/cs224w-2017/projects.html

Neo4j

https://neo4j.com/developer/python/

Past Project

https://github.com/danjizquierdo/Primary-Candidate-Analysis