Working with network data

Jim Bagrow
james.bagrow@uvm.edu
bagrow.com

Complex Networks
Winter Workshop
2019-12-16
About myself
Understanding networks from data

Community detection

**Link communities reveal multiscale complexity in networks**

Yong-Yeul Ahn\(^1,2\), James P. Bagrow\(^1,2\) & Sune Lehmann\(^1,3\)

**A Local Method for Detecting Communities**

James P. Bagrow\(^1\) and Erik M. Boltt\(^2,3\)

1. Department of Physics, Clarkson University, Potsdam, NY 13699-5820, USA.
2. Department of Math and Computer Science, Clarkson University, Potsdam, NY 13699-5815, USA.

May 16, 2006

**Communities and bottlenecks: Trees and treelike networks have high modularity**

James P. Bagrow

Department of Engineering Sciences and Applied Mathematics, Northwestern Institute on Complex Systems, Northwestern University, Evanston, Illinois 60208, USA

(Received 2 January 2012; published 15 June 2012)

Bagrow (2012)

Applied to data

**Mesoscopic Structure and Social Aspects of Human Mobility**

James P. Bagrow\(^1,2\), Yu-Ru Lin\(^3,4\)

Bagrow & Lin (2012)
**Robustness and modular structure in networks**

**JAMES P. BAGROW**
Mathematics & Statistics, University of Vermont, Burlington, VT, USA
and
Center for Complex Network Research, Northeastern University, Boston, MA, USA
(e-mail: james.bagrow@uvm.edu)

**SUÑE LEHMANN**
DTU Informatics, Technical University of Denmark, Kgs. Lyngby, Denmark
and
College of Computer and Information Science, Northeastern University, Boston, MA, USA
(e-mail: sljo@dtu.dk)

**YONG-YEOL AHN**
School of Informatics & Computing, Indiana University, Bloomington IN, USA
and
Center for Complex Network Research, Northeastern University, Boston, MA, USA
(e-mail: yyahn@indiana.edu)

---

**Abstract**

Networks with overlapping modular structure can be well modeled with a bipartite network consisting of two types of nodes representing the elements and the modules and undirected links representing which elements belong to which modules. Links are placed randomly between element and module nodes respecting these degree distributions (Newman & Park, 2003). The network consists of two types of nodes representing the elements and the modules and undirected links representing which elements belong to which modules. Links are placed randomly between element and module nodes respecting these degree distributions (Newman & Park, 2003). The network remains connected. (color online)

This causes the module network to become disconnected (bottom) even though the element network remains connected. (color online)

Elements from module A can be classified as either critical or anonymous. Critical elements are those whose removal would lead to the complete breakdown of module A. Anonymous elements are those whose removal would not affect the functionality of module A. (color online)

Critical elements are those whose removal would lead to the complete breakdown of module A. Anonymous elements are those whose removal would not affect the functionality of module A. (color online)

---

**How does missing data change the appearance of communities?**

---

Bagrow et al. (2015)
The quoter model: A paradigmatic model of the social flow of written information

James P. Bagrow1,2* and Lewis Mitchell1,2*
1Department of Mathematics and Statistics, University of Vermont, Burlington, Vermont 05405, USA
2School of Mathematical Sciences, North Terrace Campus, The University of Adelaide, Adelaide, South Australia 5005, Australia

(Received 31 October 2017; accepted 23 February 2018; published online 11 July 2018)

Measuring the flow of information between individuals

Information flow reveals prediction limits in online social activity

James P. Bagrow1,2*, Xiwei Liu1,2 and Lewis Mitchell1,2*

Modern society depends on the flow of information over online social networks, and users of popular platforms generate substantial behavioral data about themselves and their social links1,2. However, it remains unclear what fundamental limits exist when using these data to predict the activities and interests of individuals, and to what accuracy such predictions can be made using an individual’s social ties. Here, we show

postings to online social platforms present a unique opportunity to explore the temporal content of messages in conjunction with their timing, giving a richer understanding of social ties.

Information theory allows us to mathematically quantify the information contained in data and is well suited to data in the form of online written communication. Although the mathematical definition of information is somewhat distinct from our com-

(i) Alter: Hey, let’s go to the beach tomorrow.
Ego: It might rain, so let’s go to the movies.

(ii)

... word stream

\[ \lambda \]

Prob. \( q \)

\[ h_x \]

Ego

Prob. \( 1-q \)

\[ W \]

Alter

Bagrow & Mitchell (2018)
Bagrow et al (2019)
Rough Outline

- Basics
  - file formats, code, databases
- Networks from data
  - common tasks and good practices
- Case studies and examples
- Machine learning for data and networks
- Visualization (*time permitting*)
Network data are simple

- Looks like a complicated object
- Lots of measures, metrics, and algorithms to quantify and understand it
- But from a data perspective, very little to implement
Network data are simple

Store graph topology → need to define the nodes (vertices) and the links (edges):

\[ G = (V, E), |V| = N, |E| = M \]

Edgelist:

\[
\begin{align*}
&\text{Alice} & \text{Bob} \\
&\text{Bob} & \text{Carol} \\
&\text{Bob} & \text{Dani} \\
&\vdots & \vdots
\end{align*}
\]

Need identifiers for nodes and two delimiter symbols.
Network data are simple

Store graph topology → need to define the nodes (vertices) and the links (edges):

\[ G = (V, E), |V| = N, |E| = M \]

**Adjacency list:**

(ragged)

May be harder to process in some programming languages

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
<th>Carol</th>
<th>Dani</th>
<th>Bob</th>
<th>Carol</th>
<th>Erik</th>
<th>Fan</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...
Network data are simple

Store graph topology → need to define the nodes (vertices) and the links (edges):

\[ G = (V, E), |V| = N, |E| = M \]

Adjacency Matrix:

\[
\begin{pmatrix}
0 & 1 & 0 & \ldots \\
0 & 0 & 1 & \ldots \\
0 & 1 & 0 & \ldots \\
\vdots & \vdots & \vdots & \ddots
\end{pmatrix}
\]
Network data are simple

Store graph topology → need to define the nodes (vertices) and the links (edges):

\[ G = (V, E), |V| = N, |E| = M \]

Complex but more flexible

```xml
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
  http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <graph id="G" edgedefault="undirected">
    <node id="n0"/>
    <node id="n1"/>
    <node id="n2"/>
    <node id="n3"/>
    <node id="n4"/>
    <node id="n5"/>
    <node id="n6"/>
    <node id="n7"/>
    <edge source="n0" target="n2"/>
    <edge source="n1" target="n2"/>
    <edge source="n2" target="n3"/>
    <edge source="n3" target="n5"/>
    <edge source="n3" target="n4"/>
    <edge source="n4" target="n6"/>
    <edge source="n6" target="n5"/>
    <edge source="n5" target="n7"/>
  </graph>
</graphml>
What about extra attributes?

$G = (\mathcal{V}, \mathcal{E}, \mathcal{X})$

$\mathcal{X} = \text{attributes, node labels or colors, timestamps}$

Can also have edge attributes

**Edgelist**

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
<th>e1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>Carol</td>
<td>e2</td>
</tr>
<tr>
<td>Bob</td>
<td>Dani</td>
<td>e3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Node attribute list**

<table>
<thead>
<tr>
<th>Alice</th>
<th>x11</th>
<th>x12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>x21</td>
<td>x22</td>
</tr>
<tr>
<td>Carol</td>
<td>x31</td>
<td>x32</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Data surrounding network

What about **extra attributes**?

$G = (V, E, X)$

$X = \text{attributes, node labels or colors, timestamps}$

Can also have **edge attributes**

GraphML

```xml
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
    xmlns:mls="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
        http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
    <key id="d0" for="node" attr.name="color" attr.type="string">
        <default>yellow</default>
    </key>
    <key id="d1" for="edge" attr.name="weight" attr.type="double"/>
    <graph id="G" edgedefault="undirected">
        <node id="n0">
            <data key="d0">green</data>
        </node>
        <node id="n1"/>
        <node id="n2">
            <data key="d0">blue</data>
        </node>
        <node id="n3">
            <data key="d0">red</data>
        </node>
        <node id="n4"/>
        <node id="n5">
            <data key="d0">turquoise</data>
        </node>
        <edge id="e0" source="n0" target="n2">
            <data key="d1">1.0</data>
        </edge>
        <edge id="e1" source="n0" target="n1">
            <data key="d1">1.0</data>
        </edge>
    </graph>
</graphml>
```
Network data structures

To perform computations on a network, need a computable representation

```python
node2neighbors = ...

print(node2neighbors['Alice'])

print(node2neighbors['Bob'], 'Carol')
```
Network libraries

It's a good exercise to build your own data structures or even library, but in practice: lots of existing libraries

- NetworkX
- igraph
- graph-tool
- Recent versions have graph algorithms (+ always have adjacency matrix)

https://networkx.github.io
https://igraph.org
https://graph-tool.skewed.de
Graphical Interfaces and dashboards

I prefer to handle networks **computationally**, writing and running code—expressive, provenance

Interactive interfaces easier to get started but then you max out quickly!

Can be good for visualizations
Graph databases—Big Data

- neo4j
- GraphQL
- GraphDB
- SPARQL
- Jena
- Apache Giraph
- Apache Spark
- GraphX

Databases: relational, key-value, document, graph

(semantic web)

https://neo4j.com
https://jena.apache.org/
http://graphdb.ontotext.com
https://graphql.org
Graph databases—Big Data

Applications of Graph DBs:

Knowledge graphs — semantic web
Fraud detection — real time
Recommendations (Netflix, Amazon)

Graph DBs best for real-time, high-volume, local operations
Graph databases—Big Data

Applications of Graph DBs:

Knowledge graphs — semantic web
Fraud detection — real time
Recommendations (Netflix, Amazon)

Graph DBs best for real-time, high-volume, *local* operations

Knowledge Graph

Triplestore/RDF:

Courtesy: Sebastian Dery
Graph databases—Big Data

Some Knowledge Graphs

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Triples</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wikidata (2018-09-11)</td>
<td>7.2B</td>
<td>28GB</td>
</tr>
<tr>
<td>DBPedia 2016-04 English</td>
<td>1B</td>
<td>13GB</td>
</tr>
<tr>
<td>DBLP 2017</td>
<td>882M</td>
<td>1GB</td>
</tr>
<tr>
<td>Freebase</td>
<td>2B</td>
<td>11GB</td>
</tr>
<tr>
<td>YAGO2s Knowledge Base</td>
<td>159M</td>
<td>903MB</td>
</tr>
<tr>
<td>WordNet 3.1</td>
<td>5.5M</td>
<td>23MB</td>
</tr>
</tbody>
</table>

Courtesy: rdfhdt.org

Courtesy: Sebastian Dery
Network data are not simple
There is an **upstream task**

**Network data are simple**

- Looks like a complicated object
- Lots of measures, metrics, and algorithms to quantify and understand it
- But from a data perspective, very little to implement

**What defines** your network?

**Criteria for nodes?**

**Criteria for links?**

**Only simple after** addressing these questions (if you need to)
Collective Response of Human Populations to Large-Scale Emergencies
James P. Bagrow1,2,9, Dashun Wang1,2,9, Albert-László Barabási1,2,3

Link communities reveal multiscale complexity in networks
Yong-Yeol Ahn1,2,*, James P. Bagrow1,2, & Sune Lehmann1,3

Mesoscopic Structure and Social Aspects of Human Mobility
James P. Bagrow1,2, Yu-Ru Lin3,4

Example: social network from mobile phone data
Collective Response of Human Populations to Large-Scale Emergencies

James P. Bagrow1,2,*, Dashun Wang1,2,*, Albert-László Barabási1,2,3

Collective Response of Human Populations to Large-dataset, culled from the anonymized billing records of approxi-mation under external perturbations caused by emergencies. emergency, to study the real-time behavioral patterns of the local mobility [2,3,6,17] and real-time communications along the links of possibility to study such real time changes has emerged recently there is exceptional need to understand how people change their typical heterogeneity, we characterize population distributions using percentiles, proportional to the cumulative distribution.

Spatial trajectories from A in time (vertical axis), the recurrent nature of human mobility becomes evident, with a number of trips featuring both quantified by mobile phone activity), node color represents the location’s habitat detected using Infomap (see Methods), and line width discovers, which we refer to as mobility ‘habitats,’ will be shown to underlying human mobility flows; see File S1 Sec. S3 for details.
Example: social network from mobile phone data

Extracted from deidentified Call Detail Record (CDR) files

What defines your network?

Criteria for nodes?

Criteria for links?
Example: brain networks

The two metrics of a real network can be compared with those in benchmark networks such as random and regular networks. A small-world network possesses higher local interconnectivity than a random network (low clustering coefficient and short characteristic path length) and higher global integrity than a regular network (high clustering coefficient and long path lengths).

Network efficiency is a more biologically relevant metric to describe brain networks from the perspective of information flow. The global efficiency of a network is defined as the mean of the inverse of shortest path length in the network. The local efficiency of a network is measured as the averaged global efficiency of the subgraph composed of the neighbors of all nodes. Global efficiency and local efficiency measure how efficiently information is exchanged at the global and local levels, respectively.

Nodal Centrality

Several graphic metrics can be used to measure nodal centrality such as degree, efficiency, and eigenvector. These measures can quantify the roles of a node within a network from different perspectives (Fig. 2). The degree of a node is the number (in a binary graph) or the total connectivity strength (in a weighted graph) of all edges that link to the node, reflecting the most directly quantifiable measure of centrality. The nodal efficiency is calculated as the averaged reciprocal shortest path length between the node and the other nodes, representing the ability of information transfer from itself to other nodes in the entire network.

The eigenvector centrality is defined as the first eigenvector of the adjacent matrix corresponding to the largest eigenvalue, and with its recursive property, it is able to capture the global prominence of a node. In the brain networks, regions with high nodal centrality are usually referred as hubs.

Human Connectomics Based on Graph Theory

Using the abovementioned graph theory metrics, recent studies have consistently demonstrated that both human brain functional and structural networks exhibit many nontrivial topological properties such as small-worldness structure, high efficiency of information transfer, and highly connected hub regions located predominantly in the medial prefrontal and...
Example: brain networks

Network efficiency is a more biologically relevant metric to describe brain networks from the perspective of information flow. The global efficiency of a network is defined as the mean of the inverse of shortest path length in the network. The local efficiency of a network is measured as the averaged global efficiency of the subgraph composed of the neighbors of all nodes. Global efficiency and local efficiency measure how efficiently information is exchanged at the global and local levels, respectively. Using these efficiency measurements, networks with high global and local efficiencies are also considered to be small-world.

Nodal Centrality

Several graphic metrics can be used to measure nodal centrality such as degree, efficiency, and eigenvector. These measures can quantify the roles of a node within a network from different perspectives. The degree of a node is the number (in a binary graph) or the total connectivity strength (in a weighted graph) of all edges that link to the node, reflecting the most directly quantifiable measure of centrality. The nodal efficiency is calculated as the averaged reciprocal shortest path length between the node and the other nodes, representing the ability of information transfer from itself to other nodes in the entire network. The eigenvector centrality is defined as the first eigenvector of the adjacent matrix corresponding to the largest eigenvalue, and with its recursive property, it is able to capture the global prominence of a node. In the brain networks, regions with high nodal centrality are usually referred as hubs.

Human Connectomics Based on Graph Theory

Using the abovementioned graph theory metrics, recent studies have consistently demonstrated that both human brain functional and structural networks exhibit many nontrivial topological properties such as small-worldness structure, high efficiency of information transfer, and highly connected hub regions located predominantly in the medial prefrontal and...
Example: brain networks

- The two metrics of a real network can be compared with those in benchmark networks such as random and regular networks. A small-world network possesses higher local interconnectivity than a random network (low clustering coefficient and short characteristic path length) and higher global integrity than a regular network (high clustering coefficient and long path lengths).

- Network efficiency is a more biologically relevant metric to describe brain networks from the perspective of information flow. The global efficiency of a network is defined as the mean of the inverse of shortest path length in the network. The local efficiency of a network is measured as the averaged global efficiency of the subgraph composed of the neighbors of all nodes. Global efficiency and local efficiency measure how efficiently information is exchanged at the global and local levels, respectively [48, 49]. Using these efficiency measurements, networks with high global and local efficiencies are also considered to be small-world [48 – 50].

- Nodal Centrality

- Several graphic metrics can be used to measure nodal centrality such as degree, efficiency, and eigenvector. These measures can quantify the roles of a node within a network from different perspectives (Fig. 2). The degree of a node is the number (in a binary graph) or the total connectivity strength (in a weighted graph) of all edges that link to the node, reflecting the most directly quantifiable measure of centrality. The nodal efficiency is calculated as the averaged reciprocal shortest path length between the node and the other nodes, representing the ability of information transfer from itself to other nodes in the entire network [50]. The eigenvector centrality is defined as the first eigenvector of the adjacent matrix corresponding to the largest eigenvalue [51], and with its recursive property, it is able to capture the global prominence of a node [52]. In the brain networks, regions with high nodal centrality are usually referred as hubs.

- Human Connectomics Based on Graph Theory

- Using the abovementioned graph theory metrics, recent studies have consistently demonstrated that both human brain functional and structural networks exhibit many nontrivial topological properties such as small-worldness structure, high efficiency of information transfer, and highly connected hub regions located predominantly in the medial prefrontal and...
There is an upstream task

What's the best network (there may be more than one)?

Define nodes
Define edges (hyper-edges?)
Directed?
Weighted?
Use a bipartite representation or project down?

“Diseaseome”

Goh et al. PNAS (2007)
The human disease network

Do you have a bipartite network? Keep it that way?

Understanding the group dynamics and success of teams
Michael Klug¹ and James P. Bagrow¹²³

teams of collaborators

GitHub

Klug and Bagrow (2016)
Building this network from the data

GitHub provides an API that lets you access the activities (events) of users as they make changes to code, join different teams, etc.
Building this network from the data

GitHub provides an API that lets you access the activities (events) of users as they make changes to code, join different teams, etc.

```json
{
  "id": "8401895651",
  "type": "PushEvent",
  "actor": {
    "login": "bagrow",
    "display_login": "bagrow",
    "gravatar_id": "",
    "url": "https://api.github.com/users/bagrow",
  },
  "repo": {
    "id": 904212,
    "name": "bagrow/linkcomm",
    "url": "https://api.github.com/repos/bagrow/linkcomm"
  },
  "payload": {
    "action": "started"
  },
  "public": true,
  "created_at": "2018-10-11T03:33:42Z"
}
```
Building this network from the data

GitHub provides an API that lets you access the activities (events) of users as they make changes to code, join different teams, etc.

JSON data

```
{
  "id": "8401895651",
  "type": "PushEvent",
  "actor": {
    "login": "bagrow",
    "display_login": "bagrow",
    "gravatar_id": "",
    "url": "https://api.github.com/users/bagrow",
  },
  "repo": {
    "id": 904212,
    "name": "bagrow/linkcomm",
    "url": "https://api.github.com/repos/bagrow/linkcomm"
  },
  "payload": {
    "action": "started"
  },
  "public": true,
  "created_at": "2018-10-11T03:33:42Z"
}
```

https://developer.github.com/v3/
Building this network from the data

GitHub provides an API that lets you access the activities (events) of users as they make changes to code, join different teams, etc.

```json
{
  "id": "8401895651",
  "type": "PushEvent",
  "actor": {
    "login": "bagrow",
    "display_login": "bagrow",
    "avatar_url": "",
    "url": "https://api.github.com/users/bagrow",
  },
  "repo": {
    "id": 904212,
    "name": "bagrow/linkcomm",
    "url": "https://api.github.com/repos/bagrow/linkcomm"
  },
  "payload": {
    "action": "started"
  },
  "public": true,
  "created_at": "2018-10-11T03:33:42Z"
}
```
Building this network from the data

GitHub provides an API that lets you access the activities (events) of users as they make changes to code, join different teams, etc.

```
{  
  "id": "8401895651",
  "type": "PushEvent",
  "actor": {  
    "login": "bagrow",
    "display_login": "bagrow",
    "gravatar_id": "",
    "url": "https://api.github.com/users/bagrow",
  },
  "repo": {  
    "id": 904212,
    "name": "bagrow/linkcomm",
    "url": "https://api.github.com/repos/bagrow/linkcomm"
  },
  "payload": {  
    "action": "started"
  },
  "public": true,
  "created_at": "2018-10-11T03:33:42Z"
}
```
Building this network from the data

GitHub provides an API that lets you access the activities (events) of users as they make changes to code, join different teams, etc.

```json

``` Are "PushEvents" meaningful?

Are node IDs?

Insert link into bipartite graph

<table>
<thead>
<tr>
<th>Code updated</th>
<th>User</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>▼</td>
<td>▀</td>
<td>▀</td>
</tr>
</tbody>
</table>

```json


```
Building this network from the data

To build the entire network requires **scraping** their API:

- probably too slow
- API provider will probably **block you**

Solutions:

- Give up on getting the entire network and **work locally**; snowball sample?
- Find another source of data:
Building this network from the data

To build the entire network requires **scraping** their API:

- probably too slow
- API provider will probably **block you**

**Solutions:**

- Give up on getting the entire network and **work locally**; snowball sample?
- Find another source of data: [https://www.gharchive.org](https://www.gharchive.org)
Common task: thinning
Common task: thinning

Network portraits were introduced in \[22\] as a way to visualize and encode many structural properties of a given network. Specifically, the network portrait \(B\) is the array with \((\rho, k)\) elements \(B_{\rho, k}\), which counts the number of nodes who have \(k\) nodes at distance \(\rho\) for \(0 \leq \rho \leq d\), where distance is taken as the shortest path length and \(d\) is the graph's diameter. The elements of this array are computed using, e.g., Breadth-First Search. Crucially, no matter how a graph's nodes are ordered or labeled the portrait is identical. We draw several example networks and their corresponding portraits in Fig. 1. This matrix encodes many structural features of the graph. The zeroth row stores the number of nodes \(N\) in the graph: \(B_{0, k} = N\). The first row captures the degree distribution \(P(k)\): \(B_{1, k} = NP(k)\). Note that a distance \(\rho = 0\) is admissible, with two nodes \(i\) and \(j\) at distance 0 when \(i = j\). This means that the matrix \(B\) so defined has a zeroth row. It also has a zeroth column, as there may be nodes that have zero nodes at some distance \(\rho\). This occurs for nodes with eccentricity less than the graph diameter.
Common task: thinning (subsetting)

*Sometimes* necessary to remove spurious links and/or nodes

Remove singleton nodes?

Remove nodes with degree $< k$
  $\rightarrow k$-cores
Common task: thinning (subsetting)

Sometimes necessary to remove spurious links and/or nodes

Remove singleton nodes?

Remove nodes with degree < \( k \) → k-cores

💡 Temporal network?

• Keep nodes/links of a certain age
• Consider a certain time window
• But how to pick? 🤔
Common task: thinning (subsetting)

Sometimes necessary to remove spurious links and/or nodes

Remove singleton nodes?

Remove nodes with degree $< k$ → $k$-cores

Temporal network?

• Keep nodes/links of a certain age
• Consider a certain time window
• But how to pick? 😐

Choices depend on problem area, type of data, and your scientific goals
Common task: thinning

Network is very dense, lots of potentially spurious edges

How to sparsify?
Common task: thinning

Network is very dense, lots of potentially spurious edges

How to sparsify?

Threshold this matrix?

Edge \((i,j)\) exists if \(w_{ij} > \text{cutoff}\)

weighted network
Common task: thinning

Network is very dense, lots of potentially spurious edges

How to sparsify?

Threshold this matrix?

Edge \((i,j)\) exists if \(w_{ij} > \text{cutoff}\)

weighted network
Common task: thinning

Idea: Use a local threshold
Common task: thinning

Idea: Use a local threshold

Normalize weights in the neighborhood of a node:

\[ p_{ij} = \frac{w_{ij}}{\sum_j w_{ij}} \]

Extracting the multiscale backbone of complex weighted networks

M. Ángeles Serrano\textsuperscript{a,1}, Marián Boguñá\textsuperscript{b}, and Alessandro Vespignani\textsuperscript{c,d}

\[ \text{Top} = \max_{k} \sum_{i} p_{ik} \]

\[ \text{Bottom} = \max_{k} \sum_{i} p_{ik} \]

Serrano et al, PNAS (2009)
Common task: thinning

Idea: Use a local threshold

Normalize weights in the neighborhood of a node:

\[ p_{ij} = \frac{w_{ij}}{\sum_j w_{ij}} \]

Keep \((i,j)\) with statistically significant values \(p_{ij}\)

How?

Extracting the multiscale backbone of complex weighted networks

M. Ángeles Serrano\(^a,1\), Marián Boguñá\(^b\), and Alessandro Vespignani\(^c,d\)
Common task: thinning

Idea: Use a local threshold

Normalize weights in the neighborhood of a node:

\[ p_{ij} = \frac{w_{ij}}{\sum_j w_{ij}} \]

Because \( p_{ij} \) sum to 1, imagine dropping \( k_i - 1 \) points uniformly at random onto [0,1]
Common task: thinning

Idea: Use a local threshold

Normalize weights in the neighborhood of a node:

\[ p_{ij} = \frac{w_{ij}}{\sum_j w_{ij}} \]

Because \( p_{ij} \) sum to 1, imagine dropping \( k_i - 1 \) points uniformly at random onto \([0, 1]\)

Extracting the multiscale backbone of complex weighted networks

M. Ángeles Serrano\textsuperscript{a,1}, Marián Boguñá\textsuperscript{b}, and Alessandro Vespignani\textsuperscript{c,d}

This article is a PNAS Direct Submission.

The authors declare no conflict of interest.

Because \( p_{ij} \) sum to 1, imagine dropping \( k_i - 1 \) points uniformly at random onto \([0, 1]\)

Serrano et al, PNAS (2009)
Common task: thinning

Idea: Use a local threshold

Normalize weights in the neighborhood of a node:

\[ p_{ij} = \frac{w_{ij}}{\sum_j w_{ij}} \]

Extracting the multiscale backbone of complex weighted networks

M. Ángeles Serrano, Marián Boguña, and Alessandro Vespignani

What's the prob of getting a gap between points at least as big as the observed \( p_{ij} \)?

Serrano et al, PNAS (2009)
Common task: thinning

Idea: Use a local threshold

Normalize weights in the neighborhood of a node:

\[ p_{ij} = \frac{w_{ij}}{\sum_j w_{ij}} \]

Keep edges where:

\[ 1 - (k_i - 1) \int_0^{p_{ij}} (1 - x)^{k_i - 2} \, dx = (1 - p_{ij})^{k_i - 1} < \alpha \]

Exhibiting the multiscale backbone of complex weighted networks

M. Ángeles Serrano, Marián Boguña, and Alessandro Vespignani

Serrano et al, PNAS (2009)
import networkx # http://networkx.github.io

def extract_backbone(G, weights, alpha):
    keep_graph = networkx.Graph()
    for i in G:
        neighbors = G[i]
        k = len(neighbors)
        if k > 1:
            W = sum( weights[i,j] for j in neighbors )
            for j in neighbors:
                pij = 1.0*weights[i,j]/W
                if (1-pij)**(k-1) < alpha: # edge significant
                    keep_graph.add_edge(i, j)
    return keep_graph

Common task: thinning

Easy to implement! 😄
Robustness and modular structure in networks

JAMES P. BAGROW
Mathematics & Statistics, University of Vermont, Burlington, VT, USA
and
Center for Complex Network Research, Northeastern University, Boston, MA, USA
(e-mail: james.bagrow@uvm.edu)

SUNE LEHMANN
DTU Informatics, Technical University of Denmark, Kgs Lyngby, Denmark
and
College of Computer and Information Science, Northeastern University, Boston, MA, USA
(e-mail: sji@dtu.dk)

YONG-YEOL AHN
School of Informatics & Computing, Indiana University, Bloomington IN, USA
and
Center for Complex Network Research, Northeastern University, Boston, MA, USA
(e-mail: yyahn@indiana.edu)

Example where I used the method

Bagrow et al. (2015)
Robustness and modular structure in networks

JAMES P. BAGROW
Mathematics & Statistics, University of Vermont, Burlington, VT, USA
and
Center for Complex Network Research, Northeastern University, Boston, MA, USA
(e-mail: james.bagrow@uvm.edu)

SUŅE LEHMANN
DTU Informatics, Technical University of Denmark, Kgs Lyngby, Denmark
and
College of Computer and Information Science, Northeastern University, Boston, MA, USA
(e-mail: sljo@dtu.dk)

YONG-YEOL AHN
School of Informatics & Computing, Indiana University, Bloomington IN, USA
and
Center for Complex Network Research, Northeastern University, Boston, MA, USA
(e-mail: yyahn@indiana.edu)

Example where I used the method

Common task: thinning

Applied to fMRI data

Bagrow et al. (2015)
Inferring the size of the causal universe: features and fusion of causal attribution networks

Daniel Berenberg\(^1,2\) and James P. Bagrow\(^3,2,\ast\)

\(^1\)Department of Computer Science, University of Vermont, Burlington, VT, United States
\(^2\)Vermont Complex Systems Center, University of Vermont, Burlington, VT, United States
\(^3\)Department of Mathematics & Statistics, University of Vermont, Burlington, VT, United States
*Corresponding author. Email: james.bagrow@uvm.edu, Homepage: bagrow.com

December 14, 2018

Crowdsourced knowledge graphs

Case study:
Nodes are ambiguous
Knowledge graphs

\[ s_i = "anxiety" \]
\[ s_j = "sleep loss" \]
Knowledge graphs

Nodes are identified *only* by these text…
Could be ambiguous, even within one network…

Mathematical notation:

\[ s_i = "anxiety" \]

\[ s_j = "sleep loss" \]
Knowledge graphs

Nodes are identified only by these text…
Could be ambiguous, even within one network…

Berenberg & Bagrow (2018)
Knowledge graphs

\[ s_i = "anxiety" \]
\[ s_j = "sleep loss" \]

Nodes are identified only by these text…
Could be ambiguous, even within one network…

Can we combine these different networks together?

Berenberg & Bagrow (2018)
NetFUSES: Network FUision with SEmantic Similarity

Define a semantic similarity $S$ between sentences:

- $S(s_i, s_j) \leq 1$
- $S(s_i, s_i) = 1$
- $S(s_i, s_j) = S(s_j, s_i)$

Threshold $S(s_i, s_j) \geq t \quad i, j \in V_1 \cup V_2$

edges of a fusion indicator graph:

Fuse nodes using connected components

Berenberg & Bagrow (2018)
Define a semantic similarity $S$ between sentences:

- $S(s_i, s_j) \leq 1$
- $S(s_i, s_i) = 1$
- $S(s_i, s_j) = S(s_j, s_i)$

NetFUSES: Network FUision with SEmantic Similarity

Fuse nodes using connected components

How to measure semantic similarity of text?

Berenberg & Bagrow (2018)
Machine Learning

(How to measure semantic similarity of text?)
Measuring semantic similarity with neural networks

Example: image classification

using training data: labeled images

..., (JPEG,'lion'), ...
Measuring semantic similarity with neural networks

Using training data: labeled images

..., (JPEG,'lion'), ...
Measuring semantic similarity with neural networks

Example: image classification

\[ y = \sigma(w^T x) \]

using training data: labeled images

..., (JPEG, 'lion'), ...

What training data can we use for text?
“You shall know a word by the company it keeps.”

–JR Firth

Distributional Semantics
“You shall know a word by the company it keeps.”

–JR Firth

Distributional Semantics

... worlds are yours except Europa attempt no landings there ...

Turn large text corpus into collection of word-context pairs
Predict **word** from **context**

(Or predict **context** from **word**!)

*Training data*

\[
\begin{align*}
\text{input} & \quad \text{hidden layer} \quad \text{output} \\
& \\
& \\
& \\
& \\
& \\
\end{align*}
\]

\[
\begin{align*}
i_1 & \quad i_2 & \quad \vdots & \quad i_V \\
\vdots & \quad \vdots & \quad \vdots & \quad \vdots \\
\end{align*}
\]

\[
\begin{align*}
h_1 & \quad h_2 & \quad \vdots & \quad h_d \\
\vdots & \quad \vdots & \quad \vdots & \quad \vdots \\
\end{align*}
\]

\[
\begin{align*}
o_1 & \quad o_2 & \quad \vdots & \quad o_V \\
\vdots & \quad \vdots & \quad \vdots & \quad \vdots \\
\end{align*}
\]

\[
\begin{align*}
W
\end{align*}
\]

Bengio et al. JMLR (2003)
Mikolov et al. NIPS (2013)
Predict word from context

(Or predict context from word!)

**Training data**

![Diagram of neural network with input, hidden layer, and output nodes connected by the update matrix labeled $W$.]

Bengio et al. JMLR (2003)
Mikolov et al. NIPS (2013)
Predict word from context

Training data

context

input

hidden layer

output

update matrix

\[ d \]

\[ W \]

\[ \begin{align*}
  i_1 & \rightarrow h_1 \\
  i_2 & \rightarrow h_2 \\
  \vdots & \rightarrow \vdots \\
  i_V & \rightarrow h_d
\end{align*} \]

\[ \begin{align*}
  h_1 & \rightarrow o_1 \\
  h_2 & \rightarrow o_2 \\
  \vdots & \rightarrow \vdots \\
  h_d & \rightarrow o_V
\end{align*} \]

Bengio et al. JMLR (2003)
Mikolov et al. NIPS (2013)
— Each row is an $d$-dimensional word vector

vectors encode semantics

Bengio et al. JMLR (2003)
Mikolov et al. NIPS (2013)
If this sounds like SVD, you're not crazy….

\[ M \sim \log \frac{P(w, c)}{P(w)P(c)} \]

\[ M = U \Sigma V^\top \]

\[ M \approx M_d = U_d \Sigma_d V_d^\top \]

\[ W^{\text{SVD}} = U_d \Sigma_d \]

Neural network implicitly performs \textit{weighted} factorization of \( M \)

---

**Neural Word Embedding as Implicit Matrix Factorization**

Omer Levy  
Department of Computer Science  
Bar-Ilan University  
omerlevy@gmail.com

Yoav Goldberg  
Department of Computer Science  
Bar-Ilan University  
yoav.goldberg@gmail.com

Levy & Goldberg, \textit{NIPS} (2014)
Embedding words in vector spaces has taken the world by storm

word vectors, sentence vectors, *thought* vectors...

Lots of natural language processing applications including *semantic similarity*:

\[ S(s_i, s_j) = \frac{v_i \cdot v_j}{\|v_i\| \|v_j\|} \]
Embedding words in vector spaces has taken the world by storm

word vectors, sentence vectors, thought vectors...

Lots of natural language processing applications including semantic similarity:

\[ S(s_i, s_j) = \frac{v_i \cdot v_j}{\|v_i\| \|v_j\|} \]

Must be approached with caution

"[...] word vectors contain stereotypes matching those documented with the Implicit Association Test"

Caliskan et al. Science (2017)
Semantics derived automatically from language corpora contain human-like biases

Aylin Caliskan, Joanna J. Bryson, Arvind Narayanan

"[...] word vectors contain stereotypes matching those documented with the [Implicit Association Test]"

Neural language representations predict outcomes of scientific research

James P. Bagrow1,2,*, Daniel Berenberg3,2, and Joshua Bongard3,2

1Department of Mathematics & Statistics, University of Vermont, Burlington, VT, United States
2Vermont Complex Systems Center, University of Vermont, Burlington, VT, United States
3Department of Computer Science, University of Vermont, Burlington, VT, United States
*Corresponding author. Email: james.bagrow@uvm.edu, Homepage: bagrow.com

May 17, 2018

Must be approached with caution
Machine Learning for Networks
DeepWalk: Online Learning of Social Representations

Bryan Perozzi
Stony Brook University
Department of Computer Science

Rami Al-Rfou
Stony Brook University
Department of Computer Science

Steven Skiena
Stony Brook University
Department of Computer Science

{bperozzi, ralrfou, skiena}@cs.stonybrook.edu

word embedding (word2vec):

… worlds are yours except europa attempt no landings there …
word embedding (word2vec):

DeepWalk:

- Take a short random walk on the graph
- record the visited sequence of vertices
- treat vertices as words and do embedding!

Perozzi et al. KDD (2014)
DeepWalk: Online Learning of Social Representations

Bryan Perozzi
Stony Brook University
Department of Computer Science

Rami Al-Rfou
Stony Brook University
Department of Computer Science

Steven Skiena
Stony Brook University
Department of Computer Science

{bperozzi, ralrfou, skiena}@cs.stonybrook.edu

DeepWalk:

(a) Input: Karate Graph
(b) Output: Representation

Perozzi et al. KDD (2014)
Is it also a matrix factorization problem? Yes!

Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec

Jiezhong Qiu†*, Yuxiao Dong‡, Hao Ma‡, Jian Li‡, Kuansan Wang‡, and Jie Tang†
Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec

Jiezhong Qiu†*, Yuxiao Dong‡, Hao Ma‡, Jian Li‡, Kuansan Wang†, and Jie Tang†

Table 1: The matrices that are implicitly approximated and factorized by DeepWalk, LINE, PTE, and node2vec.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepWalk</td>
<td>$\log \left( \frac{1}{T} \sum_{r=1}^{T} \left( D^{-1} A \right)^r \right) D^{-1} - \log b$</td>
</tr>
<tr>
<td>LINE</td>
<td>$\log (\text{vol}(G) D^{-1} A D^{-1}) - \log b$</td>
</tr>
<tr>
<td>PTE</td>
<td>$\log \left( \begin{bmatrix} \alpha \text{vol}(G_{ww})(D_{ww}^{-1} A_{ww}(D_{ww}^{-1})) \ \beta \text{vol}(G_{dw})(D_{dw}^{-1} A_{dw}(D_{dw}^{-1})) \ \gamma \text{vol}(G_{lw})(D_{lw}^{-1} A_{lw}(D_{lw}^{-1})) \end{bmatrix} \right) - \log b$</td>
</tr>
<tr>
<td>node2vec</td>
<td>$\log \left( \frac{1}{2T} \sum_{r=1}^{T} \left( \frac{\sum_{w, u} X_{w, u} P_{c, w, u}^{r} + \sum_{c, u} X_{c, u} P_{w, c, u}^{r}}{\sum_{w, u} X_{w, u} (\sum_{c, u} X_{c, u})} \right) \right) - \log b$</td>
</tr>
</tbody>
</table>

Notations in DeepWalk and LINE are introduced below. See detailed notations for PTE and node2vec in Section 2.

— Many methods besides DeepWalk
Random walks and diffusion on networks

Naoki Masuda a,*, Mason A. Porter b,c,d, Renaud Lambiotte c

Abstract

Random walks are a fundamental concept when studying networks.

1. Introduction
2. Random walks
3. Applications
4. Multilayer networks
5. Applications
6. Respondent-driven sampling
7. Consensus probability and time of voter models
8. DeGroot model

1. Introduction
2. Random walks
3. Applications
4. Multilayer networks
5. Applications
6. Respondent-driven sampling
7. Consensus probability and time of voter models
8. DeGroot model
Random walks are a fundamental concept when studying networks.
Random walks are a fundamental concept when studying networks.
Graph neural networks

Recall: Node attribute list

<table>
<thead>
<tr>
<th></th>
<th>x11</th>
<th>x12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bob</td>
<td>x21</td>
<td>x22</td>
</tr>
<tr>
<td>Carol</td>
<td>x31</td>
<td>x32</td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
</tr>
</tbody>
</table>

$p$ features (attributes)

Supervised learning

\[ y = f(X) \]

$N \times p$ matrix of features or predictors; each row is an observation, each column is a feature.
Graph neural networks

Supervised learning

\[ y = f(X) \]

\( N \times p \) matrix of features or predictors

Each row is an observation, each column is a feature

Easy enough when observations are independent

How to incorporate the network?
neural networks

**Idea:** propagate your data through the neural network

\[ H(0) = X \]

NN: \[ H(\ell + 1) = \sigma (H(\ell)W(\ell)) \] \( \sigma \) — activation function

---

- Duvenaud *et al.*, NIPS (2015)
- Kipf *et al.*, ICLR (2017)
Graph neural networks

Idea: propagate your data through the neural network, but hit it with the graph at each layer.

\[ H(0) = X \]

NN: \[ H(\ell+1) = \sigma \left( H(\ell) W(\ell) \right) \quad \sigma \text{ — activation function} \]

GNN: \[ H(\ell+1) = \sigma \left( \tilde{A} H(\ell) W(\ell) \right) \quad \tilde{A} \text{ — preprocessed adjacency matrix} \]

e.g. Duvenaud et al. NIPS (2015)
Kipf et al. ICLR (2017)
Graph neural networks

Idea: propagate your data through the neural network, but hit it with the graph at each layer

\[ H(0) = X \]

NN: \[ H(\ell + 1) = \sigma \left( H(\ell) W(\ell) \right) \]

\[ \sigma \text{ —activation function} \]

GNN: \[ H(\ell + 1) = \sigma \left( \tilde{A} H(\ell) W(\ell) \right) \]

\[ \tilde{A} \text{ —preprocessed adjacency matrix} \]

Applications
- classifying nodes
- predicting links
- comparing networks

e.g. Duvenaud et al. NIPS (2015)
Kipf et al. ICLR (2017)
Designing visualizations
Visualization \subset Communication

Visualizations are one tool to tackle the larger problem of communicating your results
Which kind of door handle is better?
Better? Easier to open!

Designing visualizations

Which kind of door handle is better?
Designing visualizations

"Design is how it works"
–Steve Jobs
Designing visualizations

"Design is how it works"
–Steve Jobs
Designing visualizations

“Design is how it works”
–Steve Jobs

The Design of Everyday Things
Donald A. Norman
Which kind of door handle is better?

Better? Easier to open!

**Visualizations:** better = easier to understand
Designing visualizations

• Know your message
• Know your medium
• Know your audience
• Account for strengths and weaknesses of human perception
• Keep it simple

Salience to relevance

In science communication, it is critical that visual information be interpreted efficiently and correctly. The discordance between components of an image that are most noticeable and those that are most relevant or important can compromise the effectiveness of a presentation. This discrepancy can cause viewers to mistakenly pay attention to regions of the image that are not relevant. Ultimately, the misdirected attention can negatively impact comprehension.

Salience is the physical property that sets an object apart from its surroundings. It is particularly important to ensure that salience aligns with relevance in visuals used for slide presentations. In these situations, information transmission needs to be efficient because the audience member is expected to simultaneously listen and read. In contrast, unintentional and inadvertent assignment of salience can be harmful to the communicative potential of images. In the case...

Figure 2 | Discordances between salience and relevance can be harmful.
(a) The relative visibility of hues in the color scale is asymmetric, making higher values (represented by deep red) less apparent. (b) Continuously moving images can be distracting and can compromise the viewer’s ability to concentrate on other content.

In contrast, unintentional and inadvertent assignment of salience can be harmful to the communicative potential of images. In the case...

The challenge

Six months of work
↓
~ 1000 plots
↓
5-10 figures
The challenge

Six months of work
↓
~ 1000 plots
↓
5-10 figures

Hard to remember being a *beginner*
Know your message

A figure/visualization has a goal: what do you want the reader to learn?
Know your message

A figure/visualization has a **goal**: what do you want the reader to learn?

Summary sentence:

“Cancer deaths are down, but mostly due to decreased smoking rates.”
“Algorithm B converges faster than A.”
“Bats spread Ebola, not rodents.”
“The rate of text messages increased after approximately day 45.”

Build your figure(s) with this goal in mind.
Use your summary sentence to guide the kind of visualization(s) you use

http://extremepresentation.com

Zelazny, Say it with Charts, 2001
Know your medium

Print? Web? Slides?
Know your audience
Human perception

Parsing a figure or visualization requires performing **visual tasks**
Humans are **better at some tasks** and **worse at others**

<table>
<thead>
<tr>
<th>Aspect to compare</th>
<th>easiest</th>
<th>hardest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positions on a common scale</td>
<td></td>
<td>Color hue</td>
</tr>
<tr>
<td>Positions on the same but nonaligned scales</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lengths</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Angles, slopes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Volume, color saturation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Human perception

Example: Comparing areas vs. lengths
Human perception

Example: Comparing areas vs. lengths

Avoid Pie Charts!
Human perception

Perceptual biases plague even basic graphics

Cleveland & McGill (1985)
Human perception

Perceptual biases plague even basic graphics

Cleveland & McGill (1985)
Getting it right takes time

Iterate!

Readability is the most important goal!
color
Good idea to lean on existing, evidence-based palettes (*Tableau 10*) and maps (*Viridis*)

tableau.com: How we designed the new color palettes
Color blindness: the **eye** is a noisy channel

Red/Green blindness is most common → avoid it

Figure 1 | Ishihara color-vision test plate. (a) Viewers with normal color vision should see the numeral ‘6’. (b) Changing lightness of background improves contrast.

Don't rely completely on color—tweak hue/saturation to improve contrast.
Put it all together:

Keep it simple?

More Options:
- Find a song on this chart
- Just cluster similar songs
- Split View
- Show Album Cover
- Avoid Overlapping
- How to read this viz?

About the Data: The data visualized here were pulled from Spotify API. Most data attributes are computed by Spotify's audio analysis algorithms.

kenziemurphy.github.io/vinyl/
Network Visualizations

The image shows a network visualization with various interconnected nodes labeled with food items such as olive oil, zucchini, carob, Chinese cabbage, angelica, cacao, mussel, cereal, holy basil, and more. The network likely represents connections or pairings between these food items, possibly indicating flavors, nutritional similarities, or culinary applications. The diagram includes a variety of categories such as Cereal, Plants, Animal products, Flowers, Plant derivatives, Herbs, Meats, Nuts and seeds, Alcoholic beverages, Spices, and Dairy, each represented by distinct colors. The nodes are connected by lines, suggesting relationships or interactions between the items.
Before we begin, a **tough question**: is a network visualization **appropriate**?

Ghoniem *et al.* InfoVis'04 (2004)
Foucault Welles & Meirelles (2015)
Foucault Welles & Xu (2018)

Alternative approaches

Bagrow *et al.* EPL (2008)
Bagrow & Boltt (2019)
Schulman *et al.* (2011)
Network portraits

Network portraits were introduced in (Bagrow et al. 2008) and encode many structural properties of a given network. Specifically, the network portrait $B$ is the array with $(\ell, k)$ elements $B^{\ell, k}$, which is the number of nodes who have $k$ nodes at distance $\ell$ for $0 \leq \ell \leq d$ and $0 \leq k \leq N - 1$, where distance is taken as the shortest path length and $d$ is the graph's diameter. The elements of this array are computed using, e.g., Breadth-First Search. Crucially, no matter how a graph's nodes are ordered or labeled, the portrait is identical. We draw several example networks and their corresponding portraits in Fig. 1.

This matrix encodes many structural features of the graph. The zeroth row stores the number of nodes $N$: $B^{0, k} = N \delta_{k, 1}$. The first row captures the degree distribution $P(k)$: $B^{1, k} = NP(k)$, as neighbors are at distance $\ell = 1$. The second row captures the distribution of next-nearest neighbors, and so forth for higher rows. The number of edges $M$ is $\sum N k^B^{1, k} = 2M$. The graph diameter $d$ is $\max \{ \ell | B^{\ell, k} > 0 \text{ for } k > 0 \}$. The shortest path distribution is also captured: the number of shortest paths of length $\ell$ is $\frac{1}{2} \sum N k^B^{\ell, k}$. And the portrait for a random graph is different from highly ordered structures such as lattices (Fig. 1), demonstrating how dimensionality and regularity of the network is captured in the portrait (Bagrow et al. 2008).

The most important property of portraits is that they are a graph invariant.

**Network visualizations**

Before we begin, a tough question: is a network visualization appropriate?

Foucault Welles & Meirelles (2015)
Foucault Welles & Xu (2018)

**Alternative approaches**

Bagrow et al. EPL (2008)
Bagrow & Bollt (2019)
Schulman et al. (2011)
Network portraits

Network portraits were introduced in (Bagrow et al. 2008) and visualize and encode many structural properties of a given network. Specifically, the network portrait $B$ is the array with $(\ell,k)$ elements $B_{\ell,k}$ equal to the number of nodes who have $k$ nodes at distance $\ell$ for $0 \leq \ell \leq d$ and $0 \leq k \leq N-1$, where distance is taken as the shortest path length and $d$ is the graph's diameter. The element of this array is computed using, e.g., Breadth-First Search. Crucially, no matter how a graph's nodes are ordered or labeled the portrait is identical. We draw several example networks and their corresponding portraits in Fig. 1.

This matrix encodes many structural features of the graph. The zeroth row stores the number of nodes $N$ in the graph: $B_{0,k} = N \delta_{k,1}$. The first row captures the degree distribution $P(k)$: $B_{1,k} = N P(k)$, as neighbors are at distance $\ell = 1$. The second row captures the distribution of next-nearest neighbors, and so forth for higher rows. The number of edges $M$ is given by $M = \sum N_k k B_{1,k}$.

The graph diameter $d$ is $d = \max\{\ell | B_{\ell,k} > 0 \text{ for } k > 0\}$. The shortest path distribution is also captured: the number of shortest paths of length $\ell$ is $\frac{1}{2} \sum N_k k B_{\ell,k}$. And the portrait for random graphs are very different from highly ordered structures such as lattices (Fig. 1), demonstrating how dimensionality and regularity of the network is captured in the portrait (Bagrow et al. 2008).

One of the most important properties of portraits is that they are a graph invariant:

**Network visualizations**

Before we begin, a tough question: is a network visualization appropriate?

Foucault Welles & Meirelles (2015)
Foucault Welles & Xu (2018)

Alternative approaches

Bagrow et al. EPL (2008)
Bagrow & Bollt (2019)
Schulman et al. (2011)
Before we begin, a tough question: is a network visualization appropriate?

Foucault Welles & Meirelles (2015)
Foucault Welles & Xu (2018)

Alternative approaches

Bagrow et al. EPL (2008)
Bagrow & Bollt (2019)
Schulman et al. (2011)
Before we begin, a tough question: is a network visualization appropriate?

Foucault Welles & Meirelles (2015)
Foucault Welles & Xu (2018)

Alternative approaches

Bagrow et al. EPL (2008)
Bagrow & Bollt (2019)
Schulman et al. (2011)
Network visualizations

- Know your message
- Know your medium
- Know your audience
- Account for strengths and weaknesses of human perception
- Keep it simple

All these points still hold for visualizing networks
Aspects of a network visualization

1. Layout (node coordinates)
2. Node "mapper"
3. Link "mapper"
Aspects of a network visualization

0. Preprocessing
  • Project if bipartite?
  • **Thin** the network
  • Retain only subgraph(s)
  • Group nodes, network of **communities**?
  • …

1. Layout (node coordinates)
2. Node "mapper"
3. Link "mapper"
Aspects of a network visualization

0. Preprocessing
   • Project if bipartite?
   • Thin the network
   • Retain only subgraph(s)
   • Group nodes, network of communities?
   • …

1. Layout (node coordinates)
2. Node "mapper"
3. Link "mapper"

Apps
- Cytoscape
- Gephi
1. Layout (node2xy)

Place nodes in a **visually meaningful way**
Minimize link length and crossing…

*Graph drawing* — many algorithms

Can be slow for dense/large networks… should large networks even be visualized?
1. Layout (node2xy)

Place nodes in a **visually meaningful way**
Minimize link length and crossing…

*Graph drawing* — many algorithms

Can be slow for dense/large networks… should large networks even be visualized?
1. Layout (node2xy)

Tip: Algorithms are not perfect, **fine-tune by hand!** (for static visualizations)
1. Layout (node2xy)

Tip: Algorithms are not perfect, *fine-tune by hand!* (for static visualizations)
2. Node mapper (node2viz)

How to draw nodes?

Shape(s)
- Square
- Circle
- Diamond

Size(s)
- Small
- Medium
- Large

Color(s)
- Gradient from blue to red
2. Node mapper (node2viz)

How to draw nodes?

- **Shape(s)**
  - Square
  - Circle
  - Diamond

- **Size(s)**
  - Small
  - Medium
  - Large

- **Color(s)**

Tip: represent attributes by varying graphics
2. Node mapper (node2viz)

Tip: represent attributes by varying graphics

Node size ~ degree
2. Node mapper (node2viz)

Tip: represent attributes by varying graphics

Node color ~ gene expression level
3. Link mapper (link2viz)

How to draw links?

Shape(s)  |  ↓  |  ↓  |  ↓

Thickness(s)  |  |  |  |

Color(s)
3. Link mapper (link2viz)

How to draw links?

Shape(s)  ❘ ❘ ❘

Thickness(s)  ❘ ❘ ❘

Color(s)  🌈
Tip: edges don't need to be straight lines

**Edge bundling**

Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data

Danny Holten

Holton (2006)
Yong-Yeol Ahn, Sebastian Ahnert, James P. Bagrow, and A.-L. Barabási

“Flavor network and the principles of food pairing”, *Scientific Reports* 1, 196 (2011)

Flavor network. Culinary ingredients (circles) and their chemical relationship are illustrated. The color of each ingredient represents the food category that the ingredient belongs to, and the size of an ingredient is proportional to the usage frequency (selected from online recipe databases: epicureans.com, allrecipes.com, monopan.com). Two culinary ingredients are connected if they share many flavor compounds. We extracted the list of flavor compounds in each ingredient from the book "Penaz's handbook of flavor ingredients" (7th ed.) and then applied a backbone extraction method by Saito et al. [PNAS (2013) 4] (b) to pick statistically significant links between ingredients. The thickness of an edge represents the number of shared flavor compounds. To reduce clutter, edges are bundled based on the algorithm by Danny Holten (http://www.win.tue.nl/~dholten/).

Tip: edges don't need to be straight lines

Edge bundling
Tip: edges don't need to be straight lines

Edge bundling

Cytoscape:
Summary

- Basics
  - file formats, code, databases
- Networks from data
  - common tasks and good practices
- Case studies and examples
- Machine learning for data and networks
- Visualization (*time permitting*)
Challenges

• Hard to automate, generalize data analysis
• Upstream tasks defining the network
• Different fields have different needs
• Many tools, statistics, and algorithms—what to choose? Standardize?
• Gap between models and data?
• Error analysis / Uncertainty quantification

Big data:
• Gap between research and industry needs
• Graph databases—tech moving too quickly
• Visualizations (at scale)
Working with network data

Jim Bagrow
james.bagrow@uvm.edu
bagrow.com

Complex Networks
Winter Workshop
2019-12-16
Working with network data

Jim Bagrow
james.bagrow@uvm.edu
bagrow.com

Complex Networks
Winter Workshop
2019-12-16

THANK YOU