CS224W: Machine Learning with Graphs

CS224W: Machine Learning with Graphs
Jure Leskovec, Stanford University
http://cs224w.stanford.edu
Why Networks?

Networks are a general language for describing complex systems of interacting entities.
Network
The Network!
Two Types of Networks/Graphs

- **Networks (also known as Natural Graphs):**
  - **Society** is a collection of 7+ billion individuals
  - **Communication systems** link electronic devices
  - Interactions between **genes/proteins** regulate life
  - Our **thoughts** are hidden in the connections between billions of neurons in our brain

- **Information Graphs:**
  - **Information/knowledge** are organized and linked
  - **Scene graphs:** how objects in a scene relate
  - **Similarity networks:** take data, connect similar points

Sometimes the distinction is blurred
Many Types of Data are Networks

Social networks

Economic networks

Communication networks

Information networks: Web & citations

Internet

Networks of neurons

Figure 3: Higher-order cluster in the C. elegans neuronal network. A: The 4-node “bi-fan” motif, which is over-expressed in the neuronal networks. Intuitively, this motif describes a cooperative propagation of information from the nodes on the left to the nodes on the right.

B: The best higher-order cluster in the C. elegans frontal neuronal network based on the motif in (A). The cluster contains three ring motor neurons (RMEL/V/R; cyan) with many outgoing connections, serving as the source of information; six inner labial sensory neurons (IL2DL/VR/R/DR/VL; orange) with many incoming connections, serving as the destination of information; and four URA neurons (purple) acting as intermediaries. These RME neurons have been proposed as pioneers for the nerve ring, while the IL2 neurons are known regulators of nictation, and the higher-order cluster exposes their organization. The cluster also reveals that RIH serves as a critical intermediary of information processing. This neuron has incoming links from all three RME neurons, outgoing connections to five of the six IL2 neurons, and the largest total number of connections of any neuron in the cluster.

C: Illustration of the higher-order cluster in the context of the entire network. Node locations are the true two-dimensional spatial embedding of the neurons. Most information flows from left to right, and we see that RME/V/R/L and RIH serve as sources of information to the neurons on the right.
Many Types of Data are Networks

Main questions:
How are these systems organized?
What are their design properties?
Behind many systems there is an intricate wiring diagram, a network, that defines the interactions between the components.

We will never be able to model and predict these systems unless we understand the networks behind them.
Many Types of Data are Graphs

Event Graphs

Knowledge Graphs

Disease pathways

Molecules

Scene Graphs

Cell-cell similarity networks
Many Types of Data are Graphs

Main questions:
How do we take advantage of relational structure for better prediction?
Complex domains (knowledge, text, images, etc.) have rich relational structure, which can be represented as a relational graph.

By explicitly modeling relationships we achieve better performance.
But Jure, why should I care about networks?
Why Networks? Why Now?

- **Universal language for describing complex data**
  - Networks from science, nature, and technology are more similar than one would expect
- **Shared vocabulary between fields**
  - Computer Science, Social Science, Physics, Economics, Statistics, Biology
- **Data availability & computational challenges**
  - Web/mobile, bio, health, and medical
- **Impact!**
  - Social networking, Drug design, AI reasoning
Networks: Impact

- Google
- Cisco
- Facebook
- Amazon
- Pinterest
Networks and Applications
Ways to Analyze Networks

- Predict the type/color of a given node
  - Node classification
- Predict whether two nodes are linked
  - Link prediction
- Identify densely linked clusters of nodes
  - Community detection
- Measure similarity of two nodes/networks
  - Network similarity
Facebook social graph
4-degrees of separation [Backstrom-Boldi-Rosa-Ugander-Vigna, 2011]
Application: Social Circle Detection

Discover circles and why they exist

friends under the same advisor
CS department friends
college friends
‘alters’ \( v_i \)
‘ego’ \( u \)

family members
highschool friends
(2) Networks: Infrastructure

- August 15, 2003 blackout

August 14, 2003: 9:29pm EDT
20 hours before

August 15, 2003: 9:14pm EDT
7 hours after
Application: Aug 15, 2003 blackout

This reveals two important themes of this class:

- We must understand how network structure affects the robustness of a system

- Develop quantitative tools to assess the interplay between network structure and the dynamical processes on the networks, and their impact on failures

- We will learn that in reality failures follow reproducible laws, that can be quantified and even predicted using the tools of networks
(3) Networks: Knowledge

Knowledge Graphs

Heterogeneous Graphs

Multimodal Graphs
Application: Knowledge Graphs

- 300M users
- 4+B pins, 2+B boards
Application: Link Prediction

Content recommendation is link prediction
**Goal:** Map nodes to d-dimensional embeddings such that nodes with similar network neighbourhoods are embedded close together.
Example Recommendations

Query

Graph-based algorithm
Example Recommendations

Graph-based algorithm
(4) Networks: Online Media

Connections between political blogs
Polarization of the network [Adamic-Glance, 2005]
Application: Polarization on Twitter

- **Retweet networks:**
  - Polarized (left), Unpolarized (right)

Q: Is a given Wikipedia article a hoax?
- Real articles link more coherently:

Hoax article detection performance:
- 50% Random
- 66% Human
- 86% Network

Application: Predicting Virality

Information cascade in social networks
Invitation cascades: 60-90% of LinkedIn users signed up due to an invitation from another user.

Global Diffusion via Cascading Invitations: Structure, Growth, and Homophily. Anderson et al., WWW ’15.
(5) Networks: Biomedicine

Protein-protein interaction (PPI) networks:
- Nodes: Proteins
- Edges: ‘Physical’ interactions

Metabolic networks:
- Nodes: Metabolites and enzymes
- Edges: Chemical reactions
Many patients take multiple drugs to treat complex or co-existing diseases:

- 46% of people ages 70-79 take more than 5 drugs
- Many patients take more than 20 drugs to treat heart disease, depression, insomnia, etc.

**Task:** Given a pair of drugs predict adverse side effects

30% prob.

65% prob.
Application: Biomedical Graphs

- Build a heterogeneous graph
- Predict links

![Graph Diagram]

- Drug
- Protein
- $r_1$: Gastrointestinal bleed side effect
- $r_2$: Bradycardia side effect
- Drug-protein interaction
- Protein-protein interaction
Prediction Task

E.g.: How likely will Simvastatin and Ciprofloxacin, when taken together, break down muscle tissue?

Simvastatin

\(? r_2 \) (breakdown of muscle tissue)

Ciprofloxacin
Results: Side Effect Prediction

<table>
<thead>
<tr>
<th>Method</th>
<th>AUROC</th>
<th>AP@50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method (Decagon)</td>
<td>0.834</td>
<td></td>
</tr>
<tr>
<td>RESCAL Tensor Factorization [Nickel et al., ICML'11]</td>
<td>0.693</td>
<td>0.476</td>
</tr>
<tr>
<td>Multi-relational Factorization [Perros, Papalexakis et al., KDD'17]</td>
<td>0.705</td>
<td>0.567</td>
</tr>
<tr>
<td>Shallow Network Embedding [Zong et al., Bioinformatics'17]</td>
<td>0.725</td>
<td>0.643</td>
</tr>
</tbody>
</table>
About CS224W
Logistics: Teaching Staff

Instructor

Jure Leskovec

Co-Instructor

Michele Catasta

Teaching Assistants

Christina Yuan
Head TA

Lingzi (Liz) Guo

Benjamin (Ben) Hannel

Kuangcong (Cecilia) Liu

Vasco Portilheiro

Andrew Wang

Alexis Goh Weiying

Zhitaq (Rex) Ying
# Course Outline

<table>
<thead>
<tr>
<th>Date</th>
<th>1 Introduction + Structure of Graphs</th>
<th>11/7</th>
<th>14 Influence Maximization in Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>09/26</td>
<td>2 Measuring Networks, and Random Graph Model</td>
<td>11/12</td>
<td>15 Outbreak Detection in Networks</td>
</tr>
<tr>
<td>10/1</td>
<td>3 Motifs and Graphlets</td>
<td>11/14</td>
<td>16 Network Robustness and Preferential Attachment</td>
</tr>
<tr>
<td>10/3</td>
<td>4 Structural Roles in Networks</td>
<td>11/19</td>
<td>EXAM (Tue)</td>
</tr>
<tr>
<td>10/8</td>
<td>5 Spectral Clustering</td>
<td>11/21</td>
<td>17 Network Evolution</td>
</tr>
<tr>
<td>10/10</td>
<td>6 Message Passing and Node classification</td>
<td>11/26</td>
<td></td>
</tr>
<tr>
<td>10/15</td>
<td>7 Node Representation Learning</td>
<td>11/28</td>
<td></td>
</tr>
<tr>
<td>10/17</td>
<td>8 Graph Neural Networks</td>
<td>12/03</td>
<td>18 Knowledge Graphs and Metapaths</td>
</tr>
<tr>
<td>10/22</td>
<td>9 Graph Neural Networks: Hands-on</td>
<td>12/05</td>
<td>19 Network Construction, Inference and Deconvolution</td>
</tr>
<tr>
<td>10/24</td>
<td>10 Deep Generative Models for Graphs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/29</td>
<td>11 Link Analysis: PageRank and SimRank</td>
<td>method-oriented lectures</td>
<td></td>
</tr>
<tr>
<td>10/31</td>
<td>12 Network Effects and Cascading Behavior (1)</td>
<td>ML-oriented lectures</td>
<td></td>
</tr>
<tr>
<td>11/5</td>
<td>13 Network Effects and Cascading Behavior (2)</td>
<td>usecase-oriented lectures</td>
<td></td>
</tr>
</tbody>
</table>

*Thanksgiving Break*
Logistics: Website

- **http://cs224w.stanford.edu**
  - Slides posted before the class
- **Readings:**
  - Mostly research papers
- **Optional readings:**
  - Papers and pointers to additional literature
  - This will be very useful for project proposals
Logistics: Communication

- **Piazza Q&A website:**
    - Register with your @stanford.edu email
  - **Please participate and help each other!**
    - Don’t post code, annotate your questions, search for answers before you ask

- **To reach course staff (prof/TAs), always use:**
  - [cs224w-aut1920-staff@lists.stanford.edu](mailto:cs224w-aut1920-staff@lists.stanford.edu)

- We will post course announcements to Piazza (make sure you check it regularly)
Final grade will be composed of:

- **Homework: 30%**
  - Homeworks 1, 2, 3, each worth 9.6%, HW0 worth 1%
- **Exam: 30%**
- **Course project: 40%**
  - Proposal: 20%
  - Project milestone: 20%
  - Final report: 50%
  - Poster presentation: 10%
- **Extra credit: Piazza participation, code contribution**
  - Used if you are on the boundary between grades
Homework, Write-ups

- **Assignments are long and take time (10-20h)**
  - **Start early!**
  - A combination of data analysis, algorithm design, and math
  - Generally due on Thursdays 23:59 Pacific Time

- **How to submit?**
  - Upload via Gradescope ([http://gradescope.com](http://gradescope.com))
  - You will be automatically registered to Gradescope once you officially enroll in CS224W
  - Each answer must start on a new page. Read carefully the course info page!

  - **Both homework (including code) and project deliverables** must be uploaded to Gradescope!

- **Total of 2 Late Periods (LP) per student:**
  - Late period expires on Monday at 23:59 Pacific Time
  - Max 1 late period per assignment (no LP for final report)
Exam

- **November 19\(^{th}\), 2019 (in the evening)**
- **Duration: 2 hours**
- **Covers the content up to and including November 14th**
- **Open book/notes**
  - Exercises where you will have to explain the solution process
- **We will strictly enforce the Stanford Honor Code**
  - No cheating!
Course Projects

- **Course project:**
  - *Empirical analysis* of network data to develop a model of behavior
  - *Algorithms and models* to make predictions on a network dataset
  - *Scalable algorithms* for massive graphs
  - *Theoretical project* that considers a model/algorithm and derives a rigorous result about it

- **Performed in groups of up to 3 students**
  - Fine to have groups of 1 or 2. The team size will be taken under consideration when evaluating the scope of the project in breadth and depth. But 3 person teams can be more efficient.
  - Project is the *important work* for the class
  - We will help with ideas, data and mentoring
  - Start thinking about this now!
  - Ok to combine projects: Clearly indicate which part of the project is done for CS224W and which part is done for the other class.

- **Poster session:** Dec 12 12:15-3:15pm
- **Read:** [http://cs224w.stanford.edu/info.html](http://cs224w.stanford.edu/info.html)
Google Cloud infrastructure

- CS224W is generously supported by Google Cloud!
  - Each team will receive $1,500 in Google Cloud credits
  - It will allow you to work with much larger datasets compared to what you are used to with your laptop
  - You will be able to train Deep Learning models on GPUs and TPUs
  - If you never used a Cloud platform before, this is an invaluable professional experience!

- It is the first time we run the CS224W projects with such a large amount of hardware resources
  - Please be patient and understanding 😊
  - Michele will give you a Google Cloud tutorial on Friday. During the quarter, refer to him for any questions related to GCP and the project!
<table>
<thead>
<tr>
<th>Week</th>
<th>Assignment</th>
<th>Due on (11:59pm PT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Homework 0</td>
<td>Thu, October 3</td>
</tr>
<tr>
<td>3</td>
<td>Homework 1</td>
<td>Thu, October 10</td>
</tr>
<tr>
<td>4</td>
<td>Project proposal</td>
<td>Thu, October 17</td>
</tr>
<tr>
<td>5</td>
<td>Homework 2</td>
<td>Thu, October 24</td>
</tr>
<tr>
<td>7</td>
<td>Project Milestone</td>
<td>Thu, November 7</td>
</tr>
<tr>
<td>8</td>
<td>Homework 3</td>
<td>Thu, November 14</td>
</tr>
<tr>
<td>9</td>
<td>Exam</td>
<td>Thu, November 19</td>
</tr>
<tr>
<td></td>
<td>Project report</td>
<td><strong>Tue, December 10</strong> (no late periods!)</td>
</tr>
<tr>
<td></td>
<td>Poster session</td>
<td><strong>Thu, December 12</strong> 12:15-3:15pm</td>
</tr>
</tbody>
</table>
Prerequisites

- No single topic in the course is too hard by itself
- But we will cover and touch upon many topics and this is what makes the course hard

- **Good background in:**
  - Algorithms and graph theory
  - Probability and statistics
  - Linear algebra

- **Programming:**
  - You should be able to write non-trivial programs (in Python)

- **2 recitation sessions (will be recorded):**
  - SNAP.PY and Google Cloud tutorial: Skilling Auditorium, Friday 9/27, 3:00-4:20 PM
  - Review of Probability, Linear Algebra, and Proof Techniques: Skilling Auditorium, Friday 10/4, 3:00-4:20 PM
Network Analysis Tools

- We highly recommend SNAP:
  - **SNAP.PY**: Python ease of use, most of C++ scalability
    - HW0 asks you to do some very basic network analysis with snap.py
      - If you find HW0 difficult, this class is probably not for you
  - **SNAP C++**: more challenging but more scalable
  - Other tools include NetworkX, graph-tool
Starter Topic: Structure of Graphs
A network is a collection of objects where some pairs of objects are connected by links.

What is the structure of the network?
Components of a Network

- **Objects**: nodes, vertices
- **Interactions**: links, edges
- **System**: network, graph

\[ G(N,E) \]
Networks or Graphs?

- **Network** often refers to real systems
  - Web, Social network, Metabolic network
  
  **Language**: Network, node, link

- **Graph** is a mathematical representation of a network
  - Web graph, Social graph, Knowledge Graph

  **Language**: Graph, vertex, edge

We will try to make this distinction whenever it is appropriate, but in most cases we will use the two terms interchangeably.
Networks: Common Language

<table>
<thead>
<tr>
<th>Protein 1</th>
<th>Protein 2</th>
<th>Protein 5</th>
<th>Protein 9</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Actor 1</th>
<th>Actor 2</th>
<th>Actor 4</th>
<th>Actor 3</th>
</tr>
</thead>
</table>

| Movie 1   | Movie 2   | Movie 3   | |
|-----------|-----------|-----------| |

<table>
<thead>
<tr>
<th>Peter</th>
<th>Mary</th>
<th>Tom</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>friend</th>
<th>brother</th>
<th>co-worker</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Albert</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>friend</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
</table>

| |N|=4 |
| E|=4 |
If you connect individuals that work with each other, you will explore a professional network.

If you connect those that have a sexual relationship, you will be exploring sexual networks.

If you connect scientific papers that cite each other, you will be studying the citation network.

If you connect all papers with the same word in the title, what will you be exploring? It is a network, nevertheless.
How do you define a network?

- **How to build a graph:**
  - What are nodes?
  - What are edges?
- **Choice of the proper network representation of a given domain/problem determines our ability to use networks successfully:**
  - In some cases there is a unique, unambiguous representation
  - In other cases, the representation is by no means unique
  - The way you assign links will determine the nature of the question you can study
Choice of Network Representation
Directed vs. Undirected Graphs

**Undirected**
- **Links**: undirected (symmetrical, reciprocal)
- **Examples**:
  - Collaborations
  - Friendship on Facebook

**Directed**
- **Links**: directed (arcs)
- **Examples**:
  - Phone calls
  - Following on Twitter
Node Degrees

Undirected

Node degree, $k_i$: the number of edges adjacent to node $i$

$\quad k_A = 4$

Avg. degree: $\overline{k} = \langle k \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i = \frac{2E}{N}$

In directed networks we define an in-degree and out-degree. The (total) degree of a node is the sum of in- and out-degrees.

Source: Node with $k^{in} = 0$

Sink: Node with $k^{out} = 0$

$\quad k_C^{in} = 2 \quad k_C^{out} = 1 \quad k_C = 3$

$\overline{k} = \frac{E}{N}$

$k^{in} = k^{out}$
The **maximum number of edges** in an undirected graph on $N$ nodes is

$$E_{\text{max}} = \binom{N}{2} = \frac{N(N - 1)}{2}$$

An undirected graph with the number of edges $E = E_{\text{max}}$ is called a **complete graph**, and its average degree is $N - 1$
# Directedness & Average Degrees

<table>
<thead>
<tr>
<th>NETWORK</th>
<th>NODES</th>
<th>LINKS</th>
<th>DIRECTED</th>
<th>N</th>
<th>L</th>
<th>⟨k⟩</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet</td>
<td>Routers</td>
<td>Internet connections</td>
<td>Undirected</td>
<td>192,244</td>
<td>609,066</td>
<td>6.33</td>
</tr>
<tr>
<td>WWW</td>
<td>Webpages</td>
<td>Links</td>
<td>Directed</td>
<td>325,729</td>
<td>1,497,134</td>
<td>4.60</td>
</tr>
<tr>
<td>Power Grid</td>
<td>Power plants, transformers</td>
<td>Cables</td>
<td>Undirected</td>
<td>4,941</td>
<td>6,594</td>
<td>2.67</td>
</tr>
<tr>
<td>Mobile Phone Calls</td>
<td>Subscribers</td>
<td>Calls</td>
<td>Directed</td>
<td>36,595</td>
<td>91,826</td>
<td>2.51</td>
</tr>
<tr>
<td>Email</td>
<td>Email addresses</td>
<td>Emails</td>
<td>Directed</td>
<td>57,194</td>
<td>103,731</td>
<td>1.81</td>
</tr>
<tr>
<td>Science Collaboration</td>
<td>Scientists</td>
<td>Co-authorship</td>
<td>Undirected</td>
<td>23,133</td>
<td>93,439</td>
<td>8.08</td>
</tr>
<tr>
<td>Actor Network</td>
<td>Actors</td>
<td>Co-acting</td>
<td>Undirected</td>
<td>702,388</td>
<td>29,397,908</td>
<td>83.71</td>
</tr>
<tr>
<td>Citation Network</td>
<td>Paper</td>
<td>Citations</td>
<td>Directed</td>
<td>449,673</td>
<td>4,689,479</td>
<td>10.43</td>
</tr>
<tr>
<td>E. Coli Metabolism</td>
<td>Metabolites</td>
<td>Chemical reactions</td>
<td>Directed</td>
<td>1,039</td>
<td>5,802</td>
<td>5.58</td>
</tr>
<tr>
<td>Protein Interactions</td>
<td>Proteins</td>
<td>Binding interactions</td>
<td>Undirected</td>
<td>2,018</td>
<td>2,930</td>
<td>2.90</td>
</tr>
</tbody>
</table>
Bipartite graph is a graph whose nodes can be divided into two disjoint sets $U$ and $V$ such that every link connects a node in $U$ to one in $V$; that is, $U$ and $V$ are independent sets.

Examples:
- Authors-to-Papers (they authored)
- Actors-to-Movies (they appeared in)
- Users-to-Movies (they rated)
- Recipes-to-Ingredients (they contain)

“Folded” networks:
- Author collaboration networks
- Movie co-rating networks
Folded/Projected Bipartite Graphs

Projection U

- Nodes: 1, 2, 3, 4, 5, 6, 7
- Edge connections

Projection V

- Nodes: A, B, C, D
- Edge connections

U

- Nodes: 1, 2, 3, 4, 5
- Edge connections

V

- Nodes: A, B, C, D
- Edge connections
Representing Graphs: Adjacency Matrix

\[ A_{ij} = 1 \quad \text{if there is a link from node } i \text{ to node } j \]
\[ A_{ij} = 0 \quad \text{otherwise} \]

\[
A = \begin{pmatrix}
0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0
\end{pmatrix}
\]

Note that for a directed graph (right) the matrix is not symmetric.
**Adjacency Matrix**

### Undirected

- **Graph Structure:**
  - Nodes 1, 2, 3, 4
  - Connections: 1-2, 1-3, 2-3

- **Matrix Representation:**
  \[
  A_{ij} = \begin{pmatrix}
  0 & 1 & 0 & 1 \\
  1 & 0 & 0 & 1 \\
  0 & 0 & 0 & 1 \\
  1 & 1 & 1 & 0 
  \end{pmatrix}
  \]

- **Degree Formulas:**
  \[k_i = \sum_{j=1}^{N} A_{ij}\]
  \[k_j = \sum_{i=1}^{N} A_{ij}\]

- **Laplacian:**
  \[L = \frac{1}{2} \sum_{i=1}^{N} k_i = \frac{1}{2} \sum_{i,j}^{N} A_{ij}\]

### Directed

- **Graph Structure:**
  - Nodes 1, 2, 3, 4
  - Connections: 1-2, 1-3, 2-3

- **Matrix Representation:**
  \[
  A = \begin{pmatrix}
  0 & 0 & 0 & 1 \\
  1 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 \\
  0 & 1 & 1 & 0 
  \end{pmatrix}
  \]

- **Degree Formulas:**
  \[k^\text{out}_i = \sum_{j=1}^{N} A_{ij}\]
  \[k^\text{in}_j = \sum_{i=1}^{N} A_{ij}\]

- **Laplacian:**
  \[L = \sum_{i=1}^{N} k^\text{in}_i = \sum_{j=1}^{N} k^\text{out}_j = \sum_{i,j}^{N} A_{ij}\]

### Notes

- \(A_{ij} = A_{ji}\) for undirected graphs.
- \(A_{ij} \neq A_{ji}\) for directed graphs.
- \(A_{ii} = 0\) for both undirected and directed graphs.
Adjacency Matrices are Sparse
Represent graph as a set of edges:
- (2, 3)
- (2, 4)
- (3, 2)
- (3, 4)
- (4, 5)
- (5, 2)
- (5, 1)
Representing Graphs: Adjacency list

- **Adjacency list:**
  - Easier to work with if network is
    - Large
    - Sparse
  - Allows us to quickly retrieve all neighbors of a given node
    - 1:
    - 2: 3, 4
    - 3: 2, 4
    - 4: 5
    - 5: 1, 2
Most real-world networks are sparse

$E \ll E_{\text{max}}$ (or $\bar{k} \ll N-1$)

<table>
<thead>
<tr>
<th>Network Type</th>
<th>$N$</th>
<th>$\langle k \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>WWW (Stanford-Berkeley)</td>
<td>319,717</td>
<td>9.65</td>
</tr>
<tr>
<td>Social networks (LinkedIn)</td>
<td>6,946,668</td>
<td>8.87</td>
</tr>
<tr>
<td>Communication (MSN IM)</td>
<td>242,720,596</td>
<td>11.1</td>
</tr>
<tr>
<td>Coauthorships (DBLP)</td>
<td>317,080</td>
<td>6.62</td>
</tr>
<tr>
<td>Internet (AS-Skitter)</td>
<td>1,719,037</td>
<td>14.91</td>
</tr>
<tr>
<td>Roads (California)</td>
<td>1,957,027</td>
<td>2.82</td>
</tr>
<tr>
<td>Proteins (S. Cerevisiae)</td>
<td>1,870</td>
<td>2.39</td>
</tr>
</tbody>
</table>

(Source: Leskovec et al., Internet Mathematics, 2009)

Consequence: Adjacency matrix is filled with zeros!

(Density of the matrix $E/N^2$: WWW $= 1.51 \times 10^{-5}$, MSN IM $= 2.27 \times 10^{-8}$)
Edge Attributes

Possible options:

- Weight (e.g. frequency of communication)
- Ranking (best friend, second best friend...)
- Type (friend, relative, co-worker)
- Sign: Friend vs. Foe, Trust vs. Distrust
- Properties depending on the structure of the rest of the graph: number of common friends
More Types of Graphs

- **Unweighted** (undirected)

  $A_{ij} = \begin{pmatrix}
  0 & 1 & 1 & 0 \\
  1 & 0 & 1 & 1 \\
  1 & 1 & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  \end{pmatrix}$

  $A_{ii} = 0$  \quad $A_{ij} = A_{ji}

  $E = \frac{1}{2} \sum_{i,j=1}^{N} A_{ij}$  \quad $\bar{k} = \frac{2E}{N}$

  **Examples:** Friendship, Hyperlink

- **Weighted** (undirected)

  $A_{ij} = \begin{pmatrix}
  0 & 2 & 0.5 & 0 \\
  2 & 0 & 1 & 4 \\
  0.5 & 1 & 0 & 0 \\
  0 & 4 & 0 & 0 \\
  \end{pmatrix}$

  $A_{ii} = 0$  \quad $A_{ij} = A_{ji}

  $E = \frac{1}{2} \sum_{i,j=1}^{N} \text{nonzero}(A_{ij})$  \quad $\bar{k} = \frac{2E}{N}$

  **Examples:** Collaboration, Internet, Roads
More Types of Graphs

- **Self-edges (self-loops)** (undirected)
  - \( A_{ij} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \)
  - \( A_{ii} \neq 0 \)
  - \( E = \frac{1}{2} \sum_{i,j=1, i \neq j}^N A_{ij} + \sum_{i=1}^N A_{ii} \)

- **Multigraph** (undirected)
  - \( A_{ij} = \begin{pmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 3 \\ 1 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{pmatrix} \)
  - \( A_{ii} = 0 \)
  - \( E = \frac{1}{2} \sum_{i,j=1}^N \text{nonzero}(A_{ij}) \)
  - \( \bar{k} = \frac{2E}{N} \)

**Examples:** Proteins, Hyperlinks

**Examples:** Communication, Collaboration
Connectivity of Undirected Graphs

- **Connected (undirected) graph:**
  - Any two vertices can be joined by a path
  - A disconnected graph is made up by two or more connected components

**Bridge edge:** If we erase the edge, the graph becomes disconnected

**Articulation node:** If we erase the node, the graph becomes disconnected
The adjacency matrix of a network with several components can be written in a block-diagonal form, so that nonzero elements are confined to squares, with all other elements being zero:
Connectivity of Directed Graphs

- **Strongly connected directed graph**
  - has a path from each node to every other node and vice versa (e.g., A-B path and B-A path)
- **Weakly connected directed graph**
  - is connected if we disregard the edge directions

Graph on the left is connected but not strongly connected (e.g., there is no way to get from F to G by following the edge directions).
Strongly connected components (SCCs) can be identified, but not every node is part of a nontrivial strongly connected component.

In-component: nodes that can reach the SCC,
Out-component: nodes that can be reached from the SCC.
Network Representations

Email network >> directed multigraph with self-edges

Facebook friendships >> undirected, unweighted

Citation networks >> unweighted, directed, acyclic

Collaboration networks >> undirected multigraph or weighted graph

Mobile phone calls >> directed, (weighted?) multigraph

Protein Interactions >> undirected, unweighted with self-interactions
Readings