Network Science

Lecture 1: July 16, 2009 Lecturer: Maleq Khan

Scribe: Katherine Wendelsdorf

Introduction to Networks: Graph Theory I

1.1 Network Science: Definition and Relevance

1.1.1 Networks and why we study them

Definition 1 *Network*
A *Network* is a set of interconnected, interacting objects. It is a pattern of interconnections among a set of things.

Network Science: *Network Science* is the study of complex networks in order to understand their behavior and dynamics. The goal is often to determine how the structure of a network evolves over time, how structure affects propagation of information among objects and what conditions of the network's evolution allow or deter its growth.

Specifically, the aim of studying network *structure* is to predict conditions that will lead to network disintegration or failure of information propagation. For example, *What structural changes stop a disease spread across a social network? Which change prevents economic collapse in a network of bank loans?*

1.1.2 Networks as graphs

Definition 2 *Graph*
A *graph* is a mathematical representation of a set of objects that are connected when they have a specified relationship. More formally, a graph $G(V, E)$ is a set of *nodes* or *vertices*, V, that are connected by a set of *edges*, E.

Graphs can serve as useful models of interconnected objects that make up a network. Modeling a network as a specific type of graph allows one to use well studied concepts and tools of graph theory as the basis of network science.

Examples:
Social network: Nodes represent people and an edge connects a person a to another person b when one knows the other's lastname.
Contact network: Nodes represent people and an edge connects people that are in physical conact, *i.e.* they shake hands or exchange money.

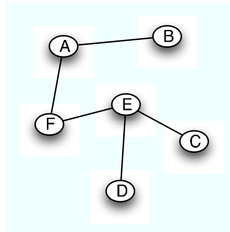

Figure 1.1: Acyclic Graph

Computer network: Nodes are processors and edges connect two computers when information is passed from one to the other.
Family Tree: nodes are members of a family and edges connected parent to child.
Economic network: Nodes are banks and edges connect one bank to another when it lends the other money.

1.2 Graph Theory: Study of Network Structure

As mentioned above, different networks may be mathematically modelled as specific types of graphs. Before describing these different types of graphs (Section 1.2.2) we define basic graph theory terminology.

1.2.1 Graph Notation and Terminology

Notation
• $G(V, E)$ denotes a graph G that is composed of a set of vertices (or nodes) V and a set of edges E.
• $V = \{A, B, C, D, E, F\}$ indicates that Graph G is composed of 6 nodes- A, B, C, D, E, F as in the case in Figure 1.1.
• $\{A, B\}$denotes an edge $e \in E$ between vertices A and B.
Hence, in the case that$\{A, B\} \in E$, there is an edge between A and B. In the case that$\{A, E\} \notin E$, there is no edge between A and B

Terminology

Incident edge: If edge $\{A, B\}$ joins nodes A and B, then edge $\{A, B\}$ is incident to A as well as B.

Degree: The degree of node A, denoted $d(A)$ is the number of incident edges to A. In Figure 1.1 $d(A) = 2$ and $d(E) = 3$.
In the case of a directed graph, the **outdegree** of a nodes is the number of edges originating

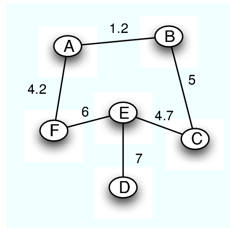

Figure 1.2: Weighted Graph

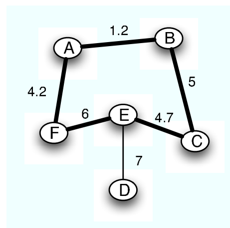

Figure 1.3: Cycle $\{A, B, C, E, F, \}$ on Graph G

from the node where as the **indegree** is the number of edges terminating to that node. In Figure 1.7 $outdegree(A) = 2$ where as $outdegree(C) = 1$. In the same graph, $indegree(A) = 0$ and $indegree(C) = 1$

Neighbors: If a pair of nodes share an incident edge, then they are called *neighbors*. If $\{A, B\} \in E$, as in Figure 1.1, then nodes A and B are neighbors of each other. $N(A)$ denotes the set of neighbors of A such that $N(A) = \{F, B\}$ in the Figure 1.1 example. **Note:** $d(A) = |N(A)|$

Weight: A quantity assigned to each edge to represent the extent of the relationship it describes. The weight of the edge $\{u, v\}$ is often written as w_{uv}

Paths and Cycles: A *path* is a sequence of vertices, $\{v_1, v_2, v_3, ..., v_i, ..., v_k\}$, such that there is an edge between consecutive vertices in the sequence, $\{v_i, v_{i+1}\}$. A **cycle** occurs when $v_1 = v_k$ as in the case of Figure 1.3.

Path Length: In the case of an unweighted graph the *path length* is the number of nodes that compose a path. In Figure 1.4 the path connecting nodes A and D that is highlighted has a path length of 4, $p(A, D) = 4$.
In that case of a weighted graph, the path length is the sum of edge weights that comprise the path. Hence, the same path in the weighted graph showin in Figure 1.4 is $p(A, D) = 4.2+6+7 = 17.2$

Parallel Edges: If two or more edges connect the same pair of nodes, these edges are called **parallel edges**. In Figure 1.5 edge $\{A, F\}$ is a parallel edge.

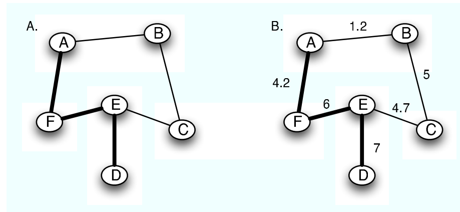

Figure 1.4: A. unweighted graph G, B. weighted graph G

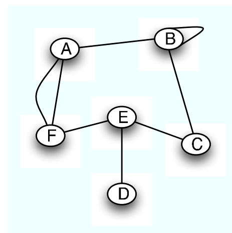

Figure 1.5: Parallel edge $\{A, F\}$ and loop $\{B, B\}$

Note: If the graph is directional and the edges are in opposite directions, the edges are not considered parallel.

Loops: An edge that links a node to itself is a **loop** as is the case for edge $\{B, B\}$ in Figure 1.5.

1.2.2 Graph Types

Below is a list of different graphs that uses terminology described in section 1.1.2 to i) define the graph type, ii) describe certain properties that are inherent to each type, and iii) suggest when they are an appropriate choice for modelling the structure of a network. Convention says that a graph is assumed simple and undirected unless explicitly stated otherwise.

Acyclic Graph: An *acyclic* graph does not possess a cycle (Figure 1.1). This type may be used when there is reason to believe that when one object A has interacted with another B, then B cannot go on to interact with others that have also interacted with A. Example: Family tree

Bi-partite Graph: A *bipartite graph* is a graph in which each node $v \in V$ belongs to one of two disjointed subsets S and U such that no edge exists between two nodes $q, v \in S$ or $q, v \in U$, but rather any edge in the graph connects a node $u \in U$ to a node $s \in S$. S and U may be referred to as **independent sets**.

Connected Graph: A graph is *connected* when each node can be reached by a path from any other node in the graph. This is always the case (but not the only case) when relationship or interaction is criteria for inclusion in the network. Example: Family tree

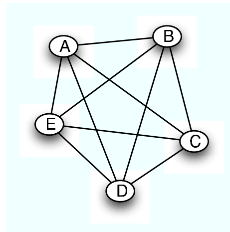

Figure 1.6: Complete Graph

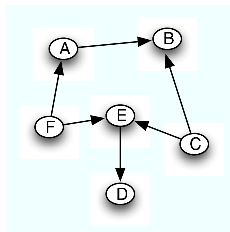

Figure 1.7: Directed Graph

Complete Graph: A graph is *complete* when there is an edge between every pair of nodes. Any node A has $V - \{A\}$ neighbors: $N(A) = V - \{A\}$ and $d(A) = n = 1$ (Figure 1.6);
Example: One may specify that in a social network all individuals in a defined space are considered in contact.

Directed Graph: In a *directed graph* information flow along an edge is unidirectional between connected nodes such that $\{A, B\} \neq \{B, A\}$(Figure 1.7);
Example: In a computer network the output of one computer,A, is the input for another computer, B.

Hypergraph: A *hypergraph* is a graph that contains *hyper edges* which is defined as a connection between > 2 nodes. An n-hypergraph specifies that an edge may connect at most n nodes. This specification is appropriate when the relationship, by definition, involves > 2 objects.
Example: A social network where each person are also linked to the location where they are connected is a 3-hypergraph. In this case a vertex set $V = \{A, B, C, D\}$ could be connected by edge set $E = \{\{A, C, D\}, \{B, C, D\}\}$ indicating that person C contacts A and B while in the same location, D

Labelled Graph: A graph is *labelled* when a symbol is associated with vertices or edges. This may be a vector representing properties of the vertex or edge (Figure **??**);
Example: Each node x may be associated with 2 values $\{x_1, x_2\}$ that are summed at time t to determine the weight of its edge at time $t + 1$.

Multigraph: A *multigraph* is a graph with parallel edges. This may be the case when edges are labelled as a type of interaction and two node interact in both ways.

Example: Two nodes representing airports are connected by an edge labelled with the airline that travels from one to the other. If multiple airlines travel between the same two airports this may be represented as a parallel edge.

Simple graph: A *simple* graph has no loop nor parallel edges.

Undirected Graph: An *undirected graph* is one in which the relationship between two connected nodes is symmetrical such that $\{A, B\} = \{B, A\}$;
Example:In a physical contact network, if person A touches person B then person B has touched person A.

Weighted Graph: In a *weighted* graph each edge of the graph is assigned a quantity describing the extent of the relationship (Figure 1.2).
Example: An edge between two people, $\{A, B\}$, in a contact network is valued as the duration of the contact. Hence, in Figure 1.2 persons A and B are in contact for 1.2 hours.