Introduction to Graph Models

BINF739 SPRING2007
Jeff Solka and Jennifer Weller

Acknowledgement

- Unless otherwise noted all figures in this lecture have been adapted from Gross and Yellen, *Graph Theory and Its Application*, Chapman and Hall/CRC Press, 2006.

What is a Graph?

- A graph consists of a collection of nodes and edges that connect the nodes.
- The nodes are entities and the edges represent relationships between the entities.
  - Nodes = proteins in a cell
  - Edges = relationships between these proteins
- Usually denoted $G = (V, E)$
  - V = vertices and E = edges
- Edges can of course be assigned weights, directions, and types

Applications of Graph Theory

- Communication networks
- Social network analysis
- Regulatory and developmental networks
- Citation networks
- Statistical data mining
  - Dimensionality reduction
  - Classification
  - Clustering

Practicalities

- We are often provided with imperfect data
- There can be errors in our edge assignments
  - False positive (relationships that appear between two nodes that are not actually there)
  - False negative (relationships that are real but were not experimentally detected)
  - Untested relationships (there could be a relationship here but there was no data to test said relationship)
- There may often be uncertainty associated with the edges.
- Uncertainty between two graphs may merely be related to the fact that in the second graph the nodes had been more extensively studied.

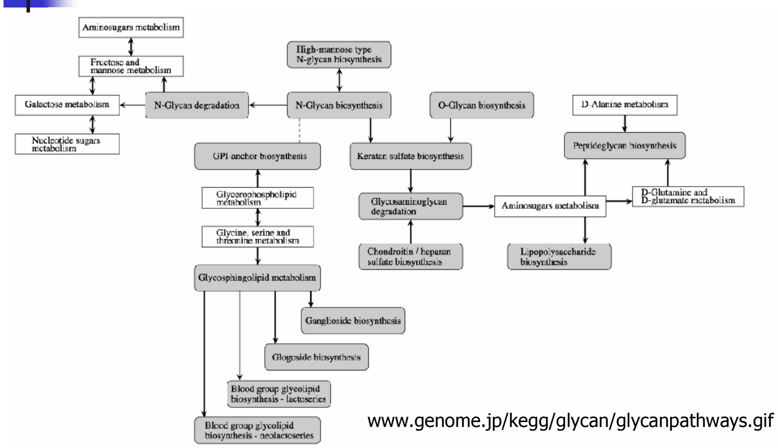
Representations of Graphs

- Graphs can have various representations and depending on the algorithm that we are implementing one representation may be more fortuitous than another.
- Edge list
- Adjacency matrix
- From to matrices

Graphs and Data Analysis

- Knowledge Representation
  - Metabolic and signal transduction networks
  - Gene Ontology (GO)
  - Bipartite graphs between genes and scientific papers that cite the genes
- Exploratory Data Analysis
  - Mapping of gene expression data onto static knowledge representation graphs
- Statistical Inference
  - Two genes are related due to frequent co-citation or that gene expression is related to protein complex co-membership
  - Random graphs such as Erdos-Reyni as well as simulation graphs that involve node permutations

Glycan Pathway as Provided by KEGG



www.genome.jp/kegg/glycan/glycanpathways.gif

Gene Ontology: A Graph of Concept Terms

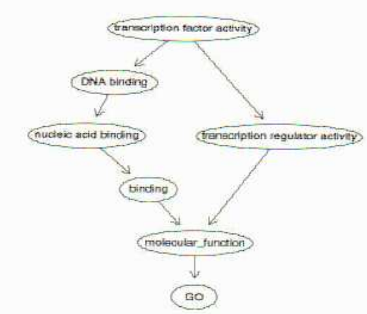


Figure 19.2. Graph of GO relationships for the term "transcription factor activity."

Gentleman et al., *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, Springer 2005.

Bipartite Gene Article Graph

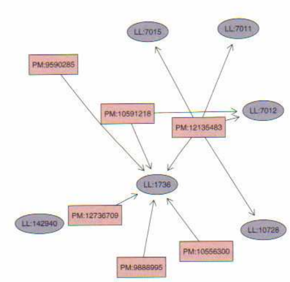


Figure 19.3. A bipartite graph between genes (LocusLink identifiers, nodes starting with "LL") and articles (PubMed identifiers, nodes starting with "PM").

Gentleman et al., *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, Springer 2005.

1.1 - Graphs and Digraphs

- Def – A <u>graph</u> $G = (V, E)$ is a mathematical structure consisting of two finite sets V and E. The elements of V are called the <u>vertices</u> (or nodes), and the elements of E are called the <u>edges</u>. Each edge has a set of one or two vertices associated with it, which are called its endpoints.

Figure 1.1.1 Line drawings of a graph A and a graph B.

Ex. 1.1.1 – The vertex and edge set of graph A is $V_A = \{p, q, r, s\}$ and $E_A = \{pq, pr, ps, rs, qs\}$

Ex. 1.1.1 – The <u>(open) neighborhood</u> of a vertex v in a graph G, denoted $N(v)$, is the set of all the neighbors of v. <u>The closed neighborhood</u> of v is given by $N[v] = N(v) \cup \{v\}$

1.1 - Simple Graphs and General Graphs

- Def. – A <u>proper edge</u> joins two distinct vertices.
- Def. – A <u>self-loop</u> is an edge that joins a single endpoint to itself.
- Def. – A <u>multi-edge</u> is a collection of two or more edges having identical end-points. The <u>edge multiplicity</u> is the number of edges within the multi-edge.
- Def. – A <u>simple graph</u> has neither self-loops nor multi-edges.
- Def. – A <u>loopless graph</u> (or multi-graph) may have multi-edges but no self-loops.
- Def. – A <u>(general) graph</u> may have self-loops and/or multi-edges.

Introduction to Graph Models
BINF739 Solka/Weller BINF739

1.1 - Null and Trivial Graphs

- Def. – A null graph is a graph whose vertex- and edge-sets are empty.
- Def. – A trivial graph is a graph consisting of one vertex and no edges.

1.1 – Edge Directions

- Def. – A directed edge (or arc) is an edge, one of whose endpoints is designated as the tail, and whose other endpoint is designated as the head.
- Def. – A directed graph (or a digraph) is a graph each of whose edges is directed.
- A digraph is simple if it has neither self-loops or multi-arcs.

1.1 – Edge Directions

Example 1.1.2: The digraph in Figure 1.1.2 is simple. Its arcs are uv, vu, and v...

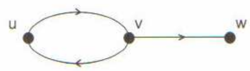


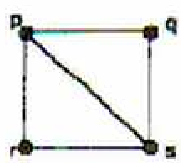
Figure 1.1.2 A simple digraph with a pair of oppositely directed arcs.

Example 1.1.3: The digraph D in Figure 1.1.3 has the graph G as its underly... graph.

Figure 1.1.3 A digraph and its underlying graph.

1.1 – Formal Specifications of Graphs and Digraphs

- Def. – A formal specification of a simple graph is given by an adjacency table with a row for each vertex, containing the list of neighbors of that vertex.

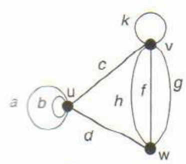


p: q r s
q: p s
r: p s
s: p q r

Figure 1.1.4 A simple graph and its formal specification.

1.1 – Formal Specifications of Graphs and Digraphs

- Def. – A <u>formal specification of a general graph</u> *G = (V, E, endpts)* consists of a list of its vertices, a list of its edges, and a two-row incidence table (specifying the endpts) function whose columns are indexed by its edges. The entries in the column corresponding to edge e are the endpoints of e. The same endpoint appears twice if e is a self-loop.



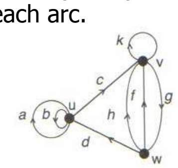
$V = \{u, v, w\}$ and $E = \{a, b, c, d, f, g, h, k\}$

edge	a	b	c	d	f	g	h	k
endpts	u	u	u	w	v	v	w	v
	u	u	v	u	w	w	v	v

Figure 1.1.5 A general graph and its formal specification.

1.1 – Formal Specifications of Graphs and Digraphs

- Def. – A formal specification of a general digraph or a mixed graph *D = (V, E, endpts, head, tail)* is obtained from the formal specifications of the underlying graph by adding the functions head : $E_G \rightarrow V_G$ and tail : $E_G \rightarrow V_G$, which designate the head vertex and tail vertex of each arc.



edge	a	b	c	d	f	g	h	k
endpts	u	u^h	u	w	v^h	v	w	v
	u^h	u	v^h	u^h	w	w^h	v^h	v^h

$$head(a) = tail(a) = head(b) = tail(b) = head(d) = tail(c) = u;$$
$$head(c) = head(h) = head(f) = tail(g) = head(k) = tail(k) = v;$$
$$head(g) = tail(d) = tail(h) = tail(f) = w.$$

Figure 1.1.6 A general digraph and its formal specification.

1.1 - Mathematical Modeling With Graphs

- A mixed graph roadmap model.

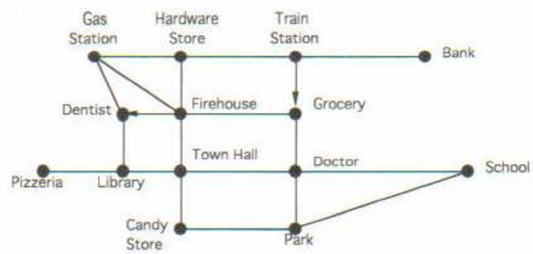


Figure 1.1.7 Road-map of landmarks in a small town.

1.1 – Mathematical Modeling with Graphs

- A digraph model of a corporate hierarchy

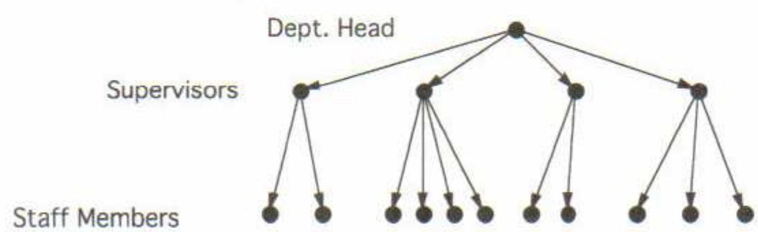


Figure 1.1.8 A corporate hierarchy.

1.1 – Degree of a Vertex

- Def. – <u>Adjacent vertices</u> are two vertices that have an endpoint in common.
- Def. – <u>Adjacent edges</u> are two edges that have an endpoint in common.
- Def. – If a vertex *v* is an endpoint of edge *e*, then *v* is said to be <u>incident</u> on *e*, and *e* is incident on *v*.
- Def. – The <u>degree (or valence)</u> of a vertex *v* in a graph *G*, denoted deg(*v*), is the number of proper edges incident on v plus twice the number of self-loops.

1.1 – Degree of a Vertex

- Def. - The <u>degree sequence</u> of a graph is the sequence formed by arranging the vertex degrees in non-increasing order.

Example 1.1.9: Figure 1.1.9 shows a graph and its degree sequence.

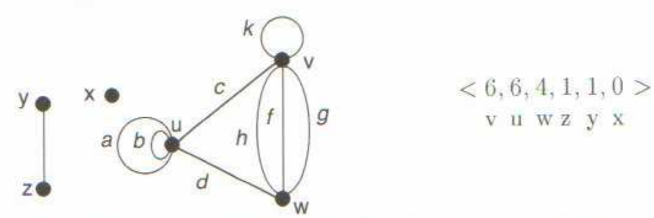


Figure 1.1.9 A graph and its degree sequence.

Introduction to Graph Models
BINF739 Solka/Weller BINF739

1.1 – Degree of a Vertex

- The degree sequence does not uniquely determine the graph

Example 1.1.10: Figure 1.1.10 shows two different graphs, G and H, with the sam[e] degree sequence.

Figure 1.1.10 Two graphs whose degree sequences are both $\langle 3,3,2,2,2,2 \rangle$.

1.1 – Degree of a Vertex

Example 1.1.11: To construct a graph whose degree sequence is $\langle 5,4,3,3,2,1,0 \rangle$, start with seven isolated vertices $v_1, v_2, \ldots, v_7$. For the even-valued terms of the sequence, draw the appropriate number of self-loops on the corresponding vertices. Thus, v_2 gets two self-loops, v_5 gets one self-loop, and v_7 remains isolated. For the four remaining odd-valued terms, group the corresponding vertices into any two pairs, for instance, v_1, v_3 and v_4, v_6. Then join each pair by a single edge and add to each vertex the appropriate number of self-loops. The resulting graph is shown in Figure 1.1.11.

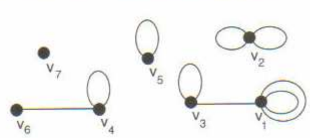
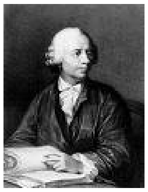


Figure 1.1.11 Constructing a graph with degree sequence $\langle 5,4,3,3,2,1,0 \rangle$.

1.1 – Degree of a Vertex

- Thm. 1.1.2 ([Euler's Degree-Sum Theorem]. The sum of the degrees of the vertices of a graph is twice the number of edges.

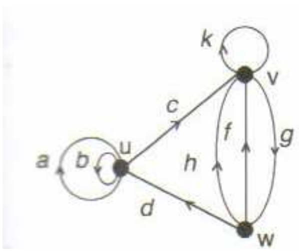


1.1 – Graphic Sequences

- Def. – A sequence <d_1, d_2, … d_n> is said to be graphic if there is a permutation of it that is the degree sequence of some simple graph. Such a simple graph is said to realize the sequence.

1.1 – Indegree and Outdegree in a Digraph

- Def. – The indegree of a vertex v in a digraph is the number of arcs directed to v; the outdegree of vertex v is the number of arcs directed from v. Each self-loop at v counts one toward the indegree of v and one toward the outdegree.



vertex	u	v	w
indegree	3	4	1
outdegree	3	2	3

1.2 Common Families of Graphs

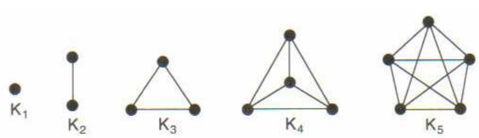


Figure 1.2.1 The first five complete graphs.

Figure 1.2.2 Two bipartite graphs.

1.2 Common Families of Graphs

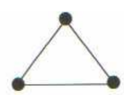


Figure 1.2.3 The smallest non-bipartite simple graph.

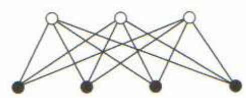


Figure 1.2.4 The complete bipartite graph $K_{3,4}$.

1.2 Common Families of Graphs

- Def. A <u>regular graph</u> is a graph whose vertices all have equal degree.

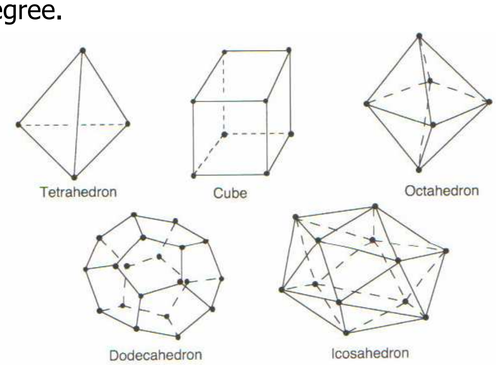


Figure 1.2.5 The five platonic graphs.

1.2 Common Families of Graphs

- The Petersen graph (a "poster child" for conjecture testing and theorem proving)

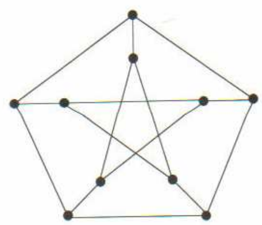


Figure 1.2.6 The Petersen graph.

Introduction to Graph Models
BINF739 Solka/Weller BINF739

1.2 Common Families of Graphs

- We can use graph theoretic models to model chemical compounds
- **Working Group on Computer-Generated Conjectures from Graph Theoretic and Chemical Databases I**
- http://dimacs.rutgers.edu/SpecialYears/2001_Data/Conjectures/abstracts.html

Figure 1.2.7 A 2-regular graph representing the oxygen molecule O_2.

Introduction to Graph Models
BINF739 Solka/Weller BINF739

1.2 Common Families of Graphs

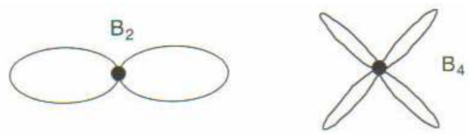


Figure 1.2.8 Bouquets B_2 and B_4.

Figure 1.2.9 Dipoles D_3 and D_4.

1.2 Common Families of Graphs

DEFINITION: A ***path graph*** P is a simple graph with $|V_P| = |E_P| + 1$ that can be drawn so that all of its vertices and edges lie on a single straight line. A path graph with n vertices and $n-1$ edges is denoted P_n.

Example 1.2.7: Path graphs P_2 and P_4 are shown in Figure 1.2.10.

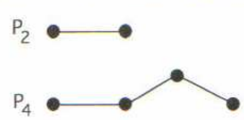


Figure 1.2.10 Path graphs P_2 and P_4.

1.2 Common Families of Graphs

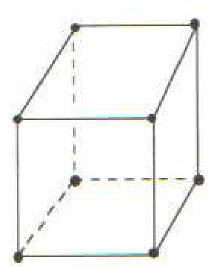
DEFINITION: A **cycle graph** is a single vertex with a self-loop or a simple graph C with $|V_C| = |E_C|$ that can be drawn so that all of its vertices and edges lie on a single circle. An n-vertex cycle graph is denoted C_n.

Example 1.2.8: The cycle graphs C_1, C_2, and C_4 are shown in Figure 1.2.11.

Figure 1.2.11 **Cycle graphs C_1, C_2, and C_4.**

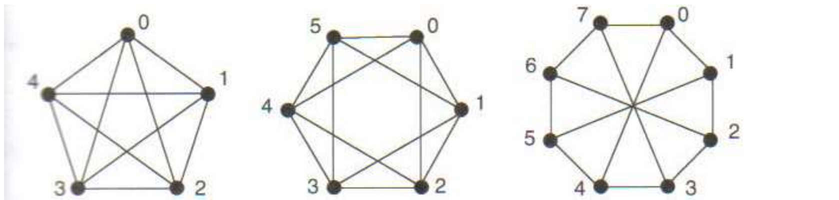
1.2 Common Families of Graphs

DEFINITION: The **hypercube graph** Q_n is the n-regular graph whose vertex-set is the set of bitstrings of length n, and such that there is an edge between two vertices if and only if they differ in exactly one bit.



Cube

1.2 Common Families of Graphs



1.2.13 **The circulant graphs** $circ(5:1,2), circ(6:1,2)$, **and** $circ(8:1,4)$.

1.2 Common Families of Graphs

DEFINITION: A simple graph G with vertex-set $V_G = \{v_1, v_2, \ldots, v_n\}$ is an ***intersection graph*** if there exists a family of sets $\mathcal{F} = \{S_1, S_2, \ldots, S_n\}$ such that vertex v_i is adjacent to v_j if and only $i \neq j$ and $S_i \cap S_j \neq \emptyset$.

DEFINITION: A simple graph is an ***interval graph*** if it is an intersection graph corresponding to a family of intervals on the real line.

Example 1.2.12: The graph in Figure 1.2.14 is an interval graph for the following family of intervals:

$a \leftrightarrow (1,3) \quad b \leftrightarrow (2,6) \quad c \leftrightarrow (5,8) \quad d \leftrightarrow (4,7)$

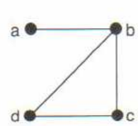


Figure 1.2.14 An interval graph.

1.2 Common Families of Graphs

- *Topics in intersection graph theory*
  - [SIAM Monographs on Discrete Mathematics and Applications #2]
    Terry A. McKee and F.R. McMorris.
    Society for Industrial and Applied Mathematics (SIAM),
    Philadelphia, PA, 1999, vii+205 pp.
  - ISBN: 0-89871-430-3
    QA 166.105.M34
- Decomposition of overlapping protein complexes: A graph theoretical method for analyzing static and dynamic protein associations Elena Zotenko1,2, Katia S Guimarães1,3, Raja Jothi1 and Teresa M Przytycka, *Algorithms for Molecular Biology* 2006, **1**:7

1.2 Common Families of Graphs

DEFINITION: The ***line graph*** $L(G)$ of a graph G has a vertex for each edge of G, and two vertices in $L(G)$ are adjacent if and only if the corresponding edges in G have a vertex in common.

Thus, the line graph $L(G)$ is the intersection graph corresponding to the endpoint sets of the edges of G.

Example 1.2.13: Figure 1.2.16 shows a graph G and its line graph $L(G)$.

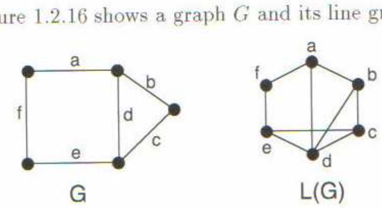
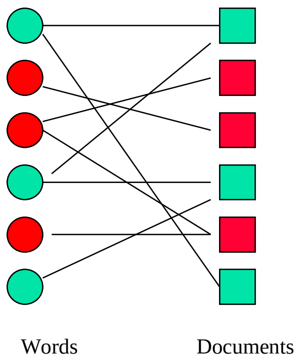


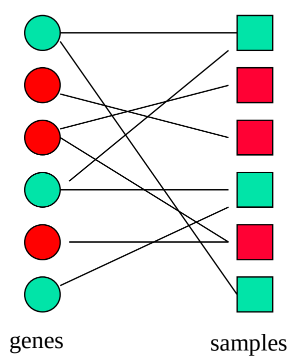
Figure 1.2.16 A graph and its line graph.

1.3 Graph Modeling Applications



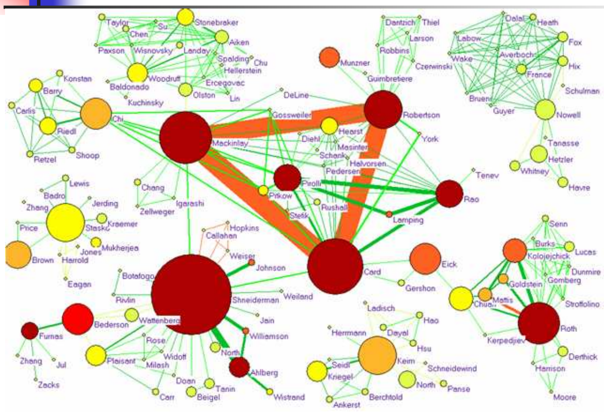
A bipartite encoding a document collection.

1.3 Graph Modeling Applications



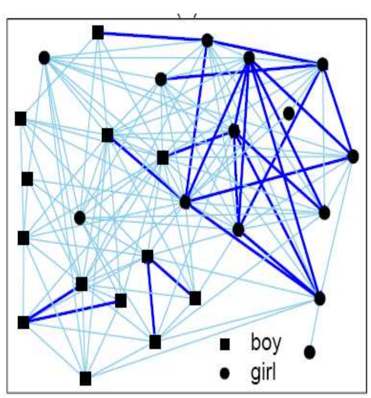
A bipartite encoding of a gene expression experiment.

1.3 Graph Modeling Applications (Evolution of co-author networks)



http://www.scimaps.org/dev/big_thumb.php?map_id=54

1.3 Graph Modeling Applications



- Classroom friendship data
- Dark lines indicate reciprocated relationships.

- **Random Effects Models for Network Data (2003)** Peter Hoff Proceedings of the National Academy of Sciences: Symposium on Social Network Analysis for National Security

1.3 Graph Modeling Applications

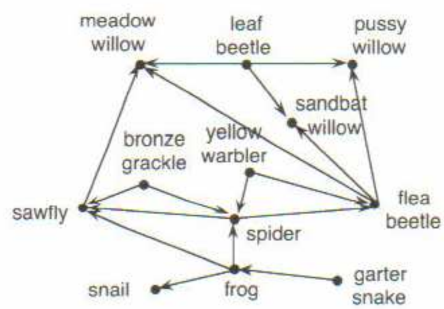
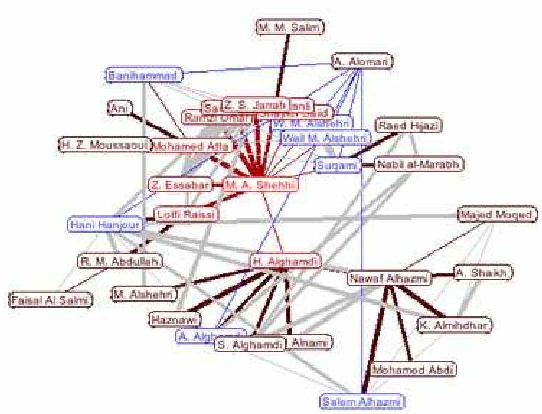


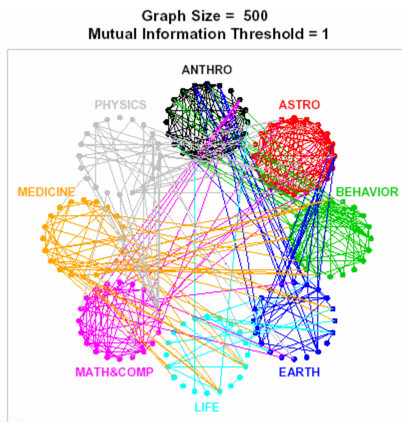
Figure 1.3.8 The food web in a Canadian willow forest.

1.3 Graph Modeling Applications



9-11 Network

1.3 – Graph Modeling Applications



nodes are documents

threshold determines which words are **important**

edge between documents that share important words

Edge between i and j if $\frac{|M_i \cap M_j|}{\sqrt{|M_i||M_j|}}$ is large

1.4 – Walks and Distance

- Def. – In a graph *G*, a <u>walk</u> from vertex v_0 to vertex v_n is an alternating sequence

$$W = \langle v_0, e_1, v_1, e_2, \ldots, v_{n-1}, e_n, v_n \rangle$$

of vertices and edges such that endpts(e_i) = $\{v_{i-1}, v_i\}$, for $i = 1, \ldots, n$. If *G* is a digraph (or a mixed graph), then *W* is a directed walk if each edge e_i is directed from vertex v_{i-1} to vertex v_i, i.e. tail(e_i) = v_{i-1} and head(e_i) = v_i

- Def. – The <u>length</u> of a walk or a directed walk is the number of edge-steps in the walk sequence.

- Def. – <u>Closed walks</u> begin and end on the same vertex, while <u>open walks</u> begin and end on different vertices.

1.4 – Walks and Distance

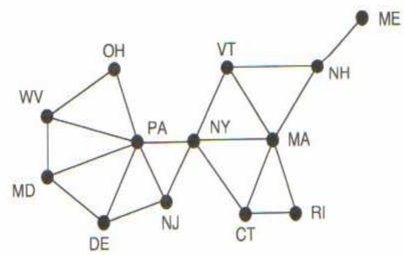


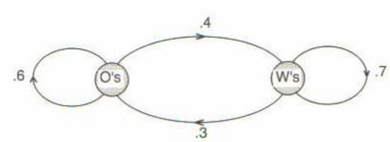
Figure 1.4.1 Geographic adjacency of the northeastern states.

1.4 – Walks and Distance

Example 1.4.2: In the Markov diagram below (from Application 1.3.12), the choice sequence of a cereal eater who buys O's, switches to W's, sticks with W's for two more boxes, and then switches back to O's is represented by the closed directed walk

$$\langle O, W, W, W, O \rangle$$

The product of the transition probabilities along a walk in any Markov diagram equals the probability that the process will follow that walk during an experimental trial. Thus, the probability that this walk occurs, when starting from O's equals $.4 \times .7 \times .7 \times .3 = 0.0588$.



	O's	W's
O's	.6	.4
W's	.3	.7

Figure 1.4.2 Markov process from Application 1.3.12.

1.4 – Walks and Distance

- Def. – The <u>distance</u> *d(s,t)* from a vertex *s* to a vertex *t* in a graph *G* is the length of a shortest s-t walk if one exists; otherwise, *d(s,t)* = infinity. For digraphs, the directed distance *d(s,t)* is the length of the shortest directed walk from *s* to *t*.

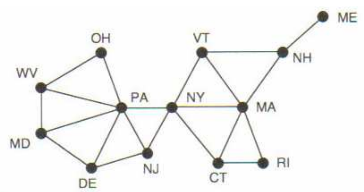


Figure 1.4.5 Geographic adjacency of the northeastern states.

1.4 – Walks and Distance

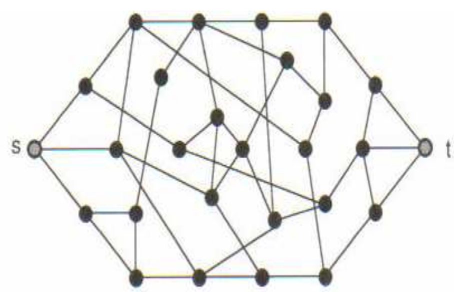


Figure 1.4.6 How might you find a shortest walk from s to t in this graph?

1.4 – Walks and Distance

- Def. – The <u>eccentricity</u> of a vertex v in a graph G, denoted *ecc(v),* is the distance from v to a vertex farthest from *v*. That is

$$ecc(v) = \max_{x \in V_G} \{d(v, x)\}$$

- The <u>diameter</u> of a graph *G*, denoted diam(*G*), is the maximum of the vertex eccentricities in *G*, or equivalently, the maximum distance between two vertices in *G*. That is,

$$diam(G) = \max_{x \in V_G} \{ecc(x)\} = \max_{x,y \in V_G} \{d(x, y)\}$$

1.4 – Walks and Distance

- Def. – The <u>radius</u> of a graph *G*, denoted *rad(G),* is the minimum of the vertex eccentricities. That is,

$$rad(G) = \min_{x \in V_G} \{ecc(x)\}$$

- Def. – A <u>central vertex</u> v of a graph *G* is a vertex with minimum eccentricity. Thus, *ecc(v) = rad(G).*

Example 1.4.7: The graph of Figure 1.4.7 below has diameter 4, achieved by the vertex pairs u, v and u, w. The vertices x and y have eccentricity 2 and all other vertices have greater eccentricity. Thus, the graph has radius 2 and central vertices x and y.

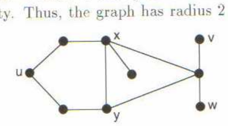


Figure 1.4.7 A graph with diameter 4 and radius 2.

1.4 – Walks and Distance

- Def. – Vertex *v* is <u>reachable</u> from vertex *u* if there is a walk from *u* to *v*.
- Def. – A graph is <u>connected</u> if for every pair of vertices *u* and *v*, there is a walk from *u* to *v*.
- Def. – A digraph is <u>connected</u> if its underlying graph is connected.

1.4 – Walks and Distance

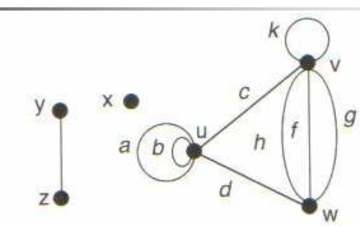


Figure 1.4.8 A non-connected graph with three components.

Introduction to Graph Models
BINF739 Solka/Weller BINF739

1.4 – Walks and Distances

- Def. – Two vertices *u* and v in a digraph D are said to be mutually reachable if D contains both a directed *u*-*v* walk and a directed *v*-u walk.
- Def. – A digraph D is strongly connected if every two of its vertices are mutually reachable.

1.4 – Walks and Distances

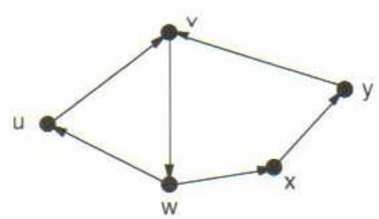
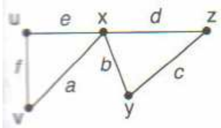


Figure 1.4.9 A strongly connected digraph.

1.5 – Paths, Cycles, and Trees

- Def. – A trail is a walk with no repeated edges.
- Def. – A path is a trail with no repeated vertices (except possibly the initial and final vertices).



$W = \langle v, a, e, f, a, d, z \rangle$
$T = \langle v, a, b, c, d, e, u \rangle$

Figure 1.5.1 Walk W is not a trail, and trail T is not a path.

1.5 – Paths, Cycles, and Trees

- Def. – A nontrivial closed path is called a cycle.
- De. – An acyclic graph is a graph that has no cycles.
- Def. – A cycle that includes every vertex of a graph is called a hamilton cycle.
- Def. – A hamilton graph is a graph that has a hamilton cycle.

1.5 – Paths, Cycles, and Trees

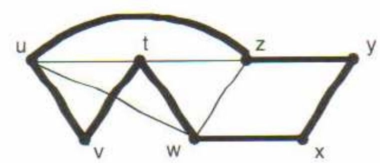


Figure 1.5.3 A hamiltonian graph.

Introduction to Graph Models
BINF739 Solka/Weller BINF739

1.5 – Paths, Cycles, and Trees

- Def. – An eulerian trail in a graph is a trail that contains every edge of that graph.
- Def. – An eulerian tour is a closed eulerian trail.
- Def. – An eulerian graph is a graph that has an eulerian tour.

Introduction to Graph Models
BINF739 Solka/Weller BINF739

1.5 – Paths, Cycles, and Trees

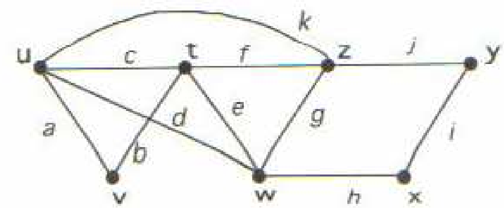


Figure 1.5.6 An eulerian graph.

1.5 – Paths, Cycles, and Trees

- Def. – The girth of a graph G with at least one cycle is the length of a shortest cycle in G. The girth if an acyclic graph is undefined.

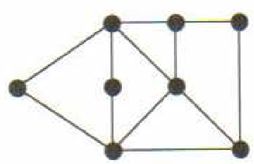
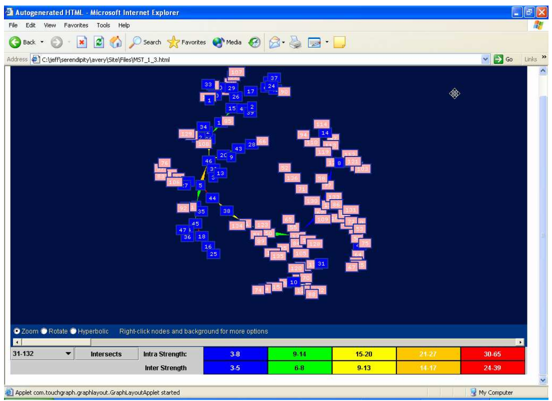


Figure 1.5.7 A graph with girth 3.

1.5 – Paths, Cycles, and Trees

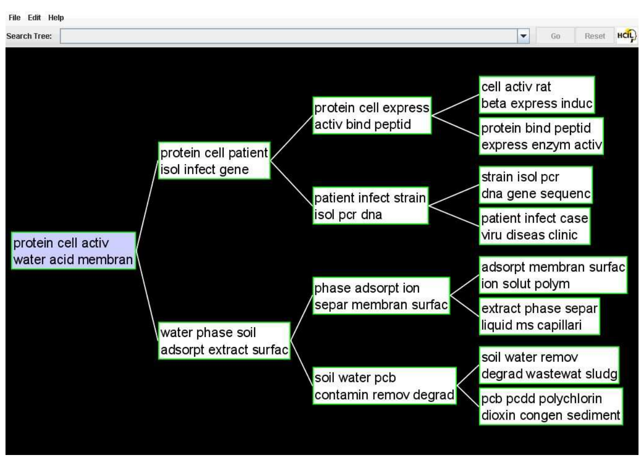
- Def. – A tree is a connected graph that has no cycles.



Introduction to Graph Models
BINF739 Solka/Weller BINF739

1.5 – Paths, Cycles, and Trees

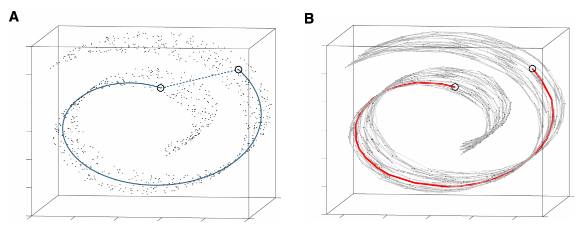
Tree



Introduction to Graph Models
BINF739 Solka/Weller BINF739

1.6 – Vertex and Edge Attributes: More Applications

- Def. – A <u>weighted graph</u> is a graph in which each edge is assigned a number, called the edge weight.



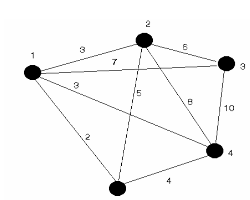
Shortest Path

R^3 Geodesic and Manifold Geodesic

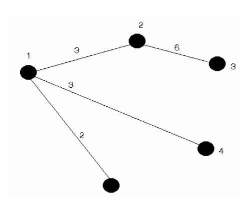
ISOMAP Geodesic and Associated Nearest Neighbor Graph

1.6 – Vertex and Edge Attributes: More Applications

- Definition (Minimal Spanning Tree (MST)) – The collection of edges that join all of the points in a set together, with the minimum possible sum of edge values. The edge values that will be used here is the distance measures stored in our interpoint distance matrix.



A graph.



Associated MST.

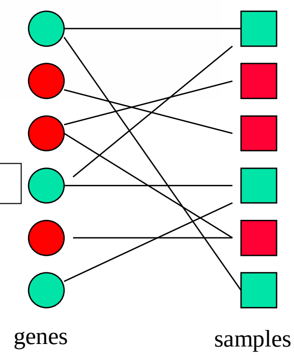
1.6 – Vertex and Edge Attributes: More Applications

Graph Partitioning

Given a graph $G = (\mathcal{V}, E)$, the classical graph bipartitioning or bisection problem is to find nearly equally-sized vertex subsets $\mathcal{V}_1^*, \mathcal{V}_2^*$ of $\mathcal{V}$ such that

$$\mathrm{cut}(\mathcal{V}_1^*, \mathcal{V}_2^*) = \min_{\mathcal{V}_1, \mathcal{V}_2} \mathrm{cut}(\mathcal{V}_1, \mathcal{V}_2).$$

The graph partitioning problem is known to be NP-complete.



1.6 – Vertex and Edge Attributes: More Applications

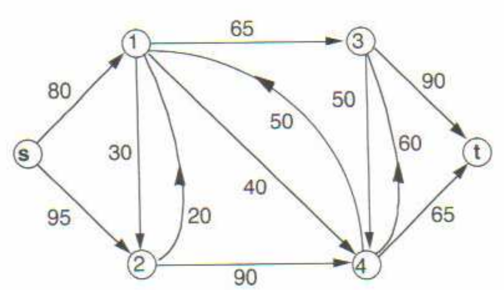


Figure 1.6.4 A maximum-flow problem.