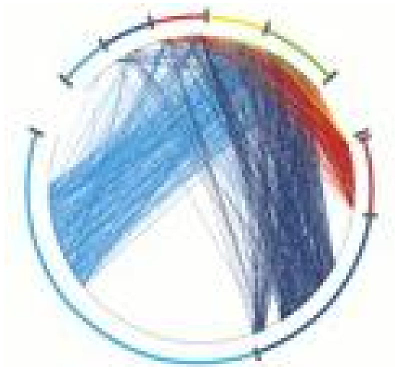
Models and Algorithms for Complex Networks

Introduction and Background

Lecture 1



Welcome!

- Introductions
  - My name in finnish: Panajotis Tsaparas
    - I am from Greece
    - I graduated from University of Toronto
      - Web searching and Link Analysis
    - In University of Helsinki for the past 2 years
  - Tutor: Evimaria Terzi
    - also Greek
- Knowledge of Greek is not required

Course overview

- The course goal
  - To read some recent and interesting papers on information networks
  - Understand the underlying techniques
  - Think about interesting problems
- Prerequisites:
  - Mathematical background on discrete math, graph theory, probabilities, linear algebra
  - The course will be more "theoretical", but your project may be more "practical"
- Style
  - Both slides and blackboard

Topics

- Measuring Real Networks
- Models for networks
- Scale Free and Small World networks
- Distributed hashing and Peer-to-Peer search
- The Web graph
  - Web crawling, searching and ranking
- Biological networks
- Gossip and Epidemics
- Graph Clustering
- Other special topics

Homework

- Two or three assignments of the following three types
  - Reaction paper
  - Problem Set
  - Presentation
- Project: Select your favorite network/algorithm/model and
  - do an experimental analysis
  - do a theoretical analysis
  - do a in-depth survey
- No final exam
- Final Grade: 50% assignments, 50% project (or 60%,40%)
- Tutorials: will be arranged on demand

Web page

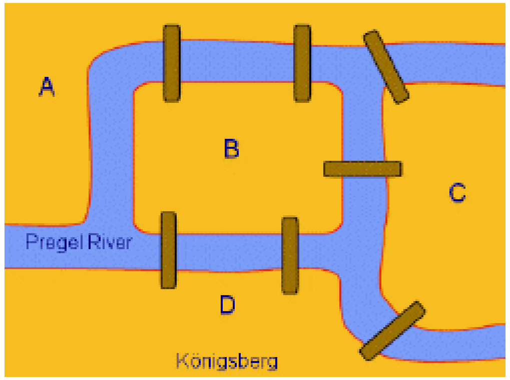
http://www.cs.helsinki.fi/u/tsaparas/MACN2006/

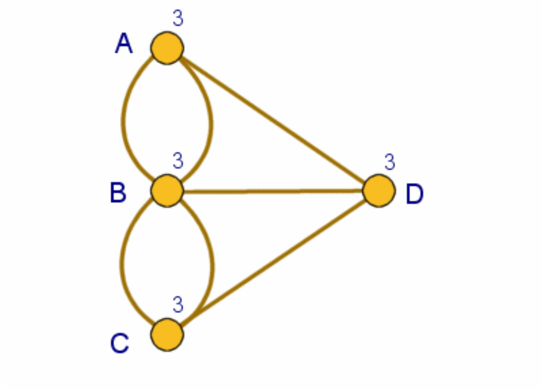
What is a network?

- Network: a collection of entities that are interconnected with links.
  - people that are friends
  - computers that are interconnected
  - web pages that point to each other
  - proteins that interact

Graphs

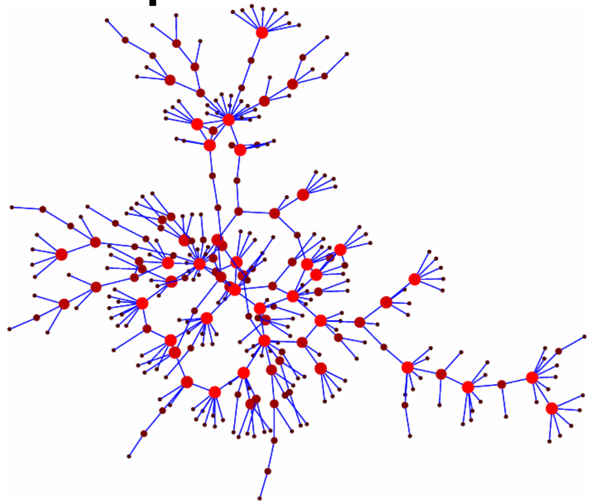
- In mathematics, networks are called graphs, the entities are nodes, and the links are edges
- Graph theory starts in the 18th century, with Leonhard Euler
  - The problem of Königsberg bridges
  - Since then graphs have been studied extensively.





Networks in the past

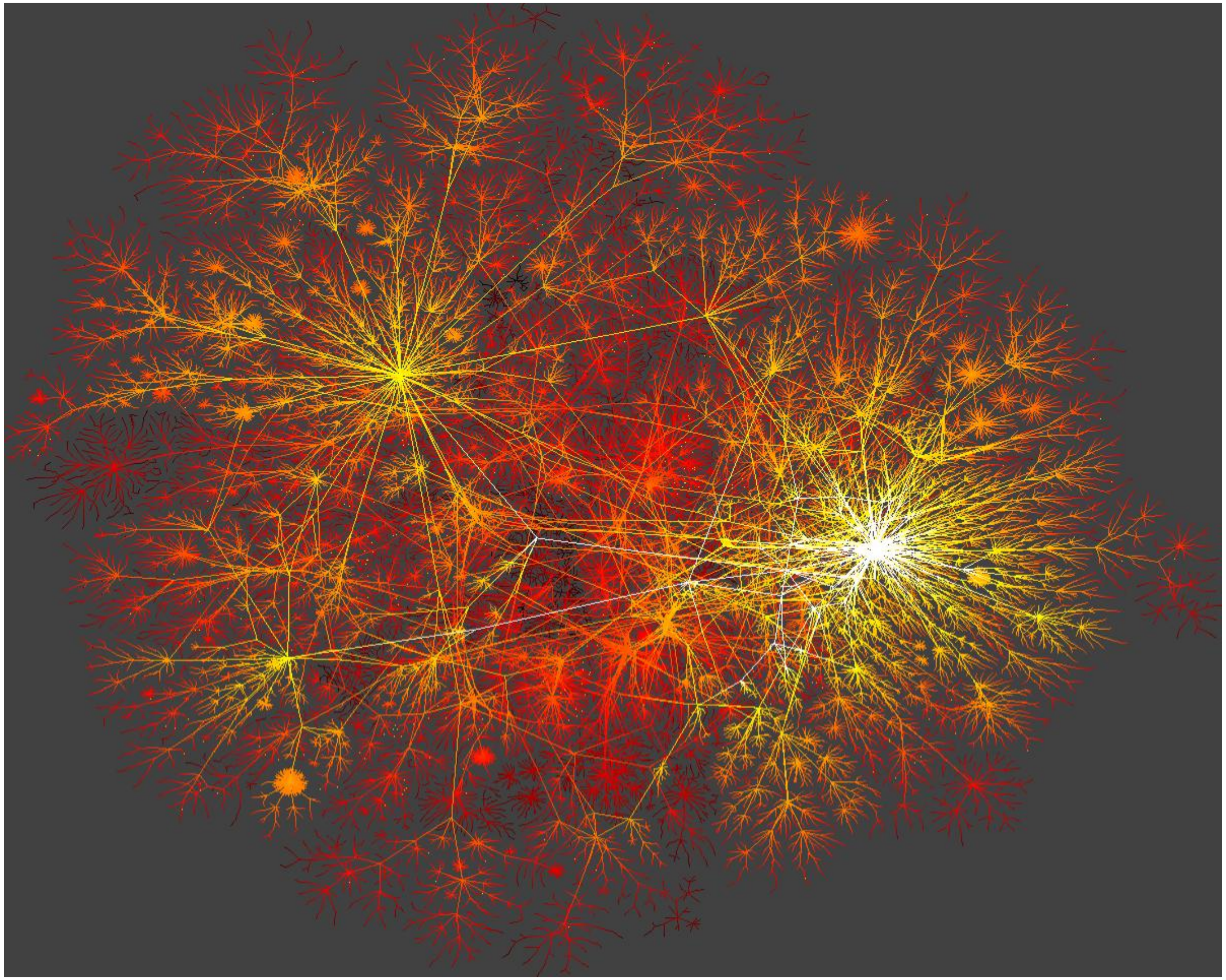
- Graphs have been used in the past to model existing networks (e.g., networks of highways, social networks)
  - usually these networks were small
  - network can be studied visual inspection can reveal a lot of information



Networks now

- More and larger networks appear
  - Products of technological advancement
    - e.g., Internet, Web
  - Result of our ability to collect more, better, and more complex data
    - e.g., gene regulatory networks
- Networks of thousands, millions, or billions of nodes
  - impossible to visualize

The internet map



Understanding large graphs

- What are the statistics of real life networks?
- Can we explain how the networks were generated?

Measuring network properties

- Around 1999
  - Watts and Strogatz, Dynamics and small-world phenomenon
  - Faloutsos[3], On power-law relationships of the Internet Topology
  - Kleinberg et al., The Web as a graph
  - Barabasi and Albert, The emergence of scaling in real networks

Real network properties

- Most nodes have only a small number of neighbors (degree), but there are some nodes with very high degree (power-law degree distribution)
  - scale-free networks
- If a node x is connected to y and z, then y and z are likely to be connected
  - high clustering coefficient
- Most nodes are just a few edges away on average.
  - small world networks
- Networks from very diverse areas (from internet to biological networks) have similar properties
  - Is it possible that there is a unifying underlying generative process?

Generating random graphs

- Classic graph theory model (Erdös-Renyi)
  - each edge is generated independently with probability p
- Very well studied model but:
  - most vertices have about the same degree
  - the probability of two nodes being linked is independent of whether they share a neighbor
  - the average paths are short

Modeling real networks

- Real life networks are not “random”
- Can we define a model that generates graphs with statistical properties similar to those in real life?
  - a flurry of models for random graphs

Processes on networks

- Why is it important to understand the structure of networks?

- Epidemiology: Viruses propagate much faster in scale-free networks
- Vaccination of random nodes does not work, but targeted vaccination is very effective

Web search

- First generation search engines: the Web as a collection of documents
  - Suffered from spammers, poor, unstructured, unsupervised content, increase in Web size
- Second generation search engines: the Web as a network
  - use the anchor text of links for annotation
  - good pages should be pointed to by many pages
  - good pages should be pointed to by many good pages
    - PageRank algorithm, Google!

The future of networks

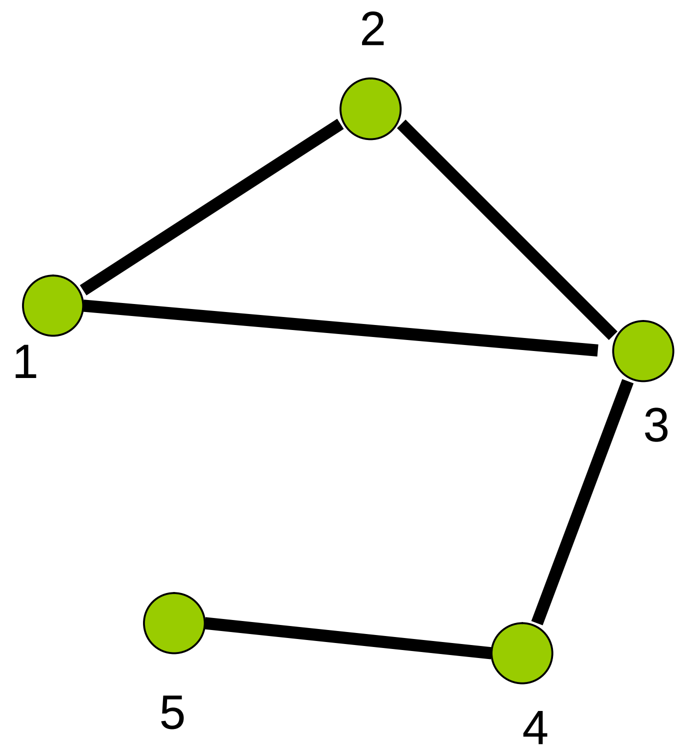
- Networks seem to be here to stay
  - More and more systems are modeled as networks
  - Scientists from various disciplines are working on networks (physicists, computer scientists, mathematicians, biologists, sociologist, economists)
  - There are many questions to understand.

Mathematical Tools

- Graph theory
- Probability theory
- Linear Algebra

Graph Theory

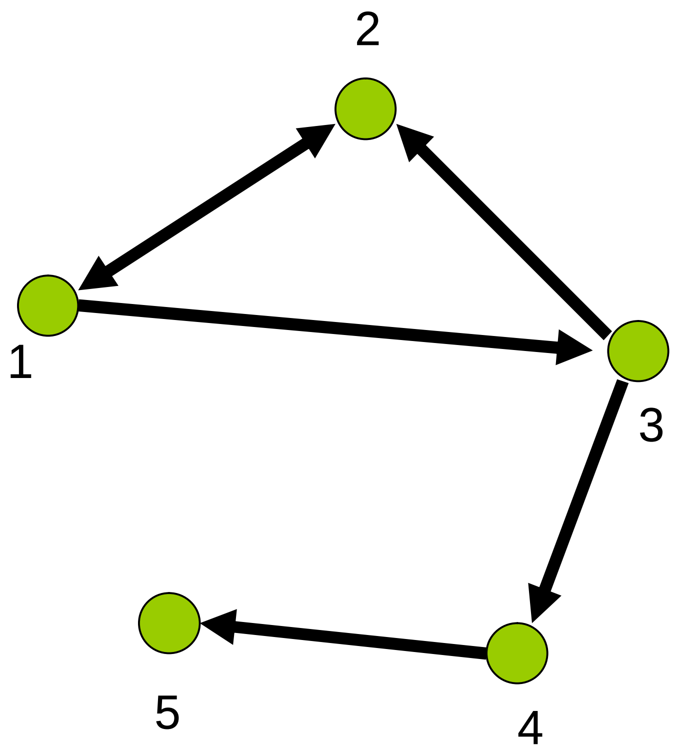
- Graph G=(V,E)
  - V = set of vertices
  - E = set of edges



undirected graph
E={(1,2),(1,3),(2,3),(3,4),(4,5)}

Graph Theory

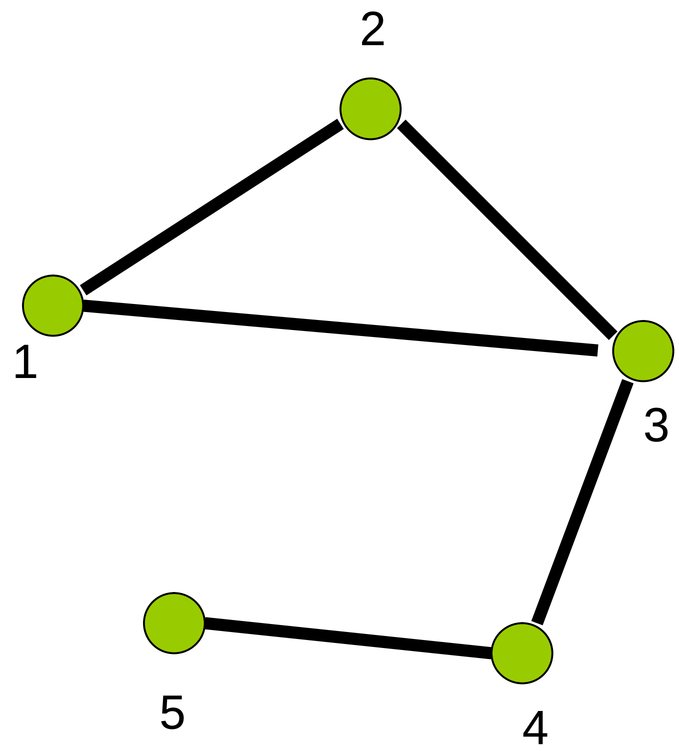
- Graph G=(V,E)
  - V = set of vertices
  - E = set of edges

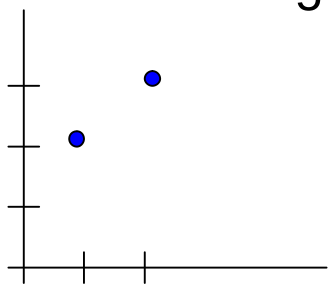


directed graph
E={‹1,2›, ‹2,1› ‹1,3›, ‹3,2›, ‹3,4›, ‹4,5›}

Undirected graph

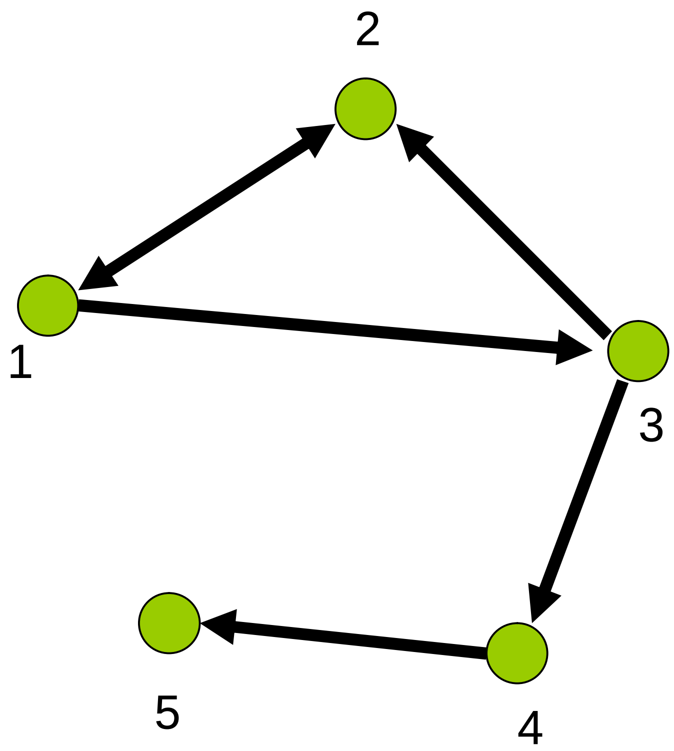
- degree d(i) of node i
  - number of edges incident on node i
- degree sequence
  - [d(i),d(2),d(3),d(4),d(5)]
  - [2,2,2,1,1]
- degree distribution
  - [(1,2),(2,3)]





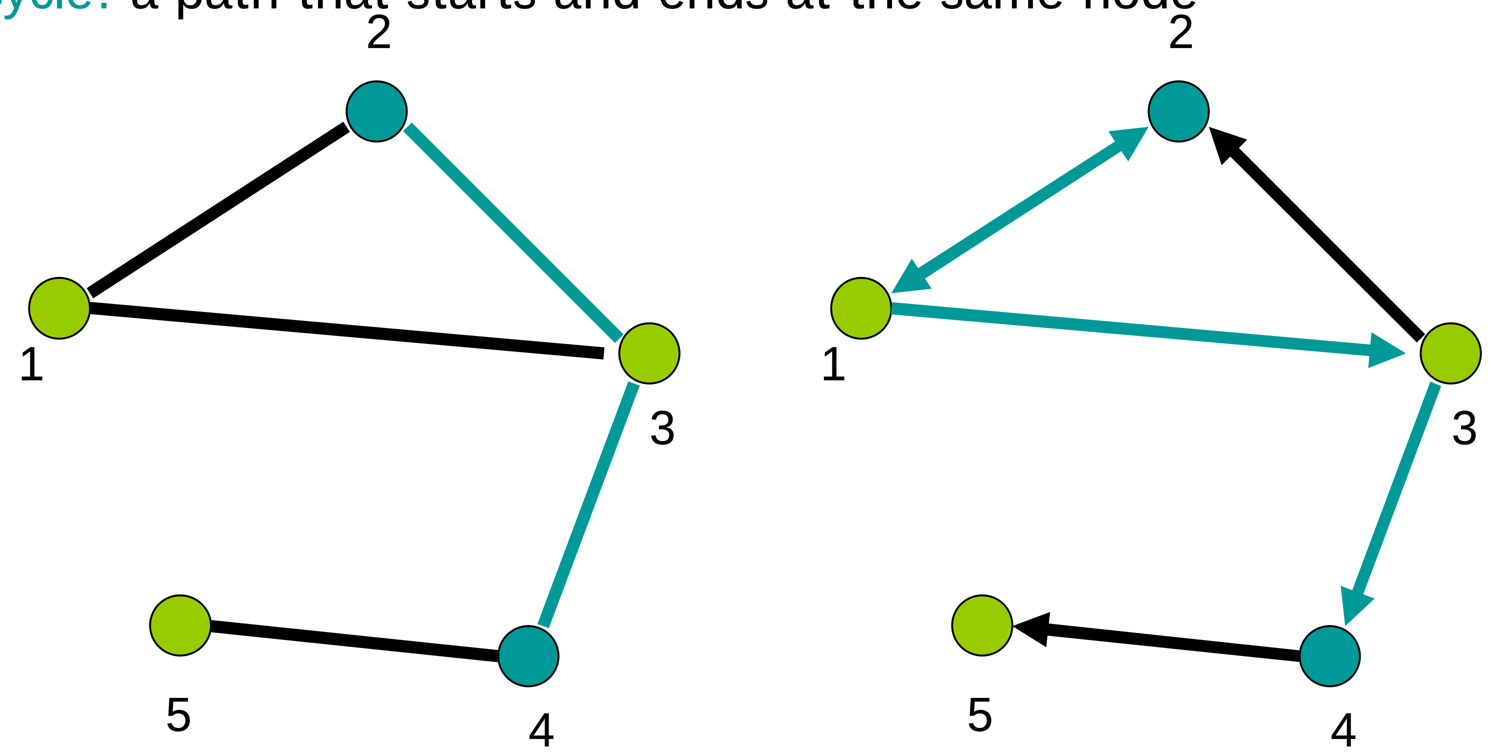
Directed Graph

- in-degree $d_{in}(i)$ of node i
  - number of edges pointing to node i
- out-degree $d_{out}(i)$ of node i
  - number of edges leaving node i
- in-degree sequence
  - [1,2,1,1,1]
- out-degree sequence
  - [2,1,2,1,0]



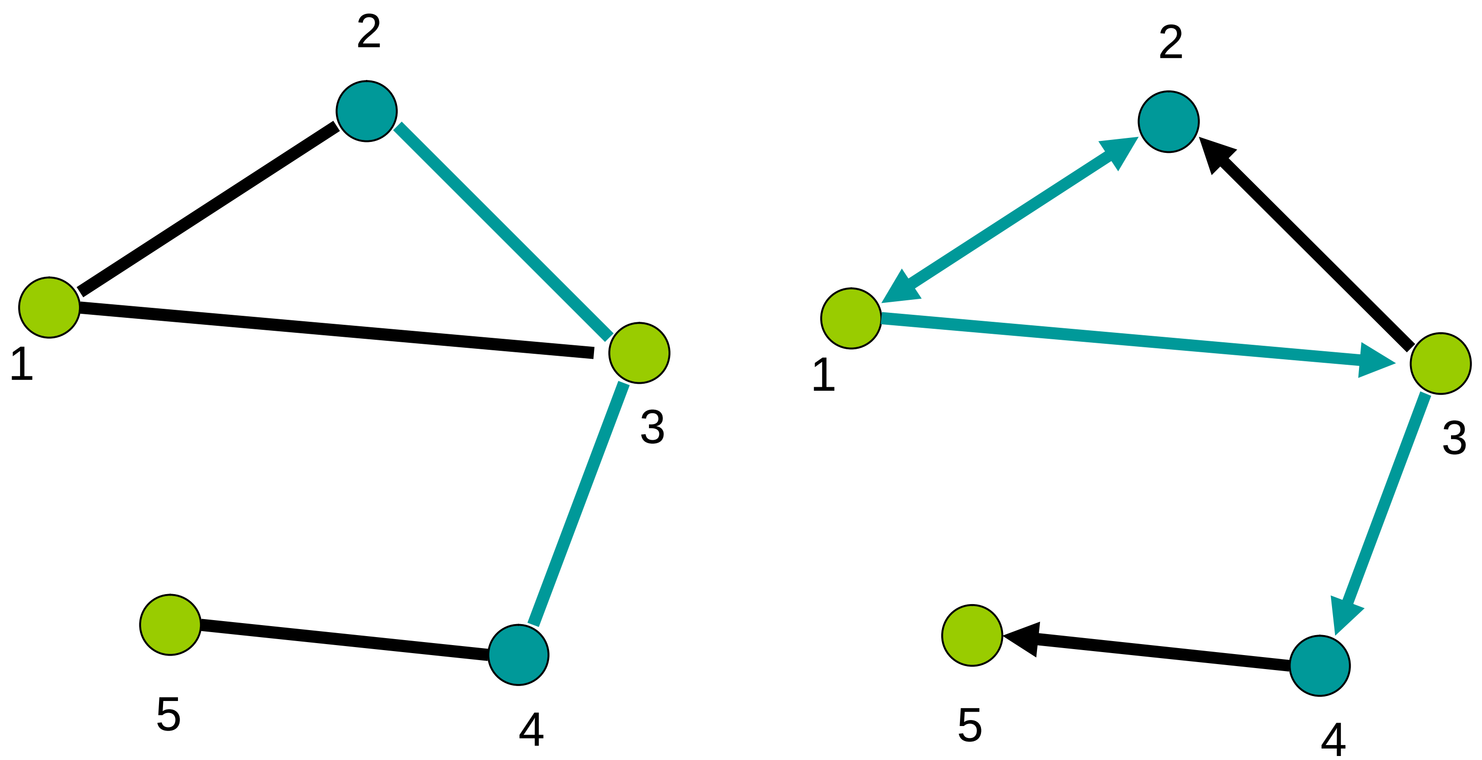
Paths

- Path from node i to node j: a sequence of edges (directed or undirected from node i to node j)
  - path length: number of edges on the path
  - nodes i and j are connected
  - cycle: a path that starts and ends at the same node



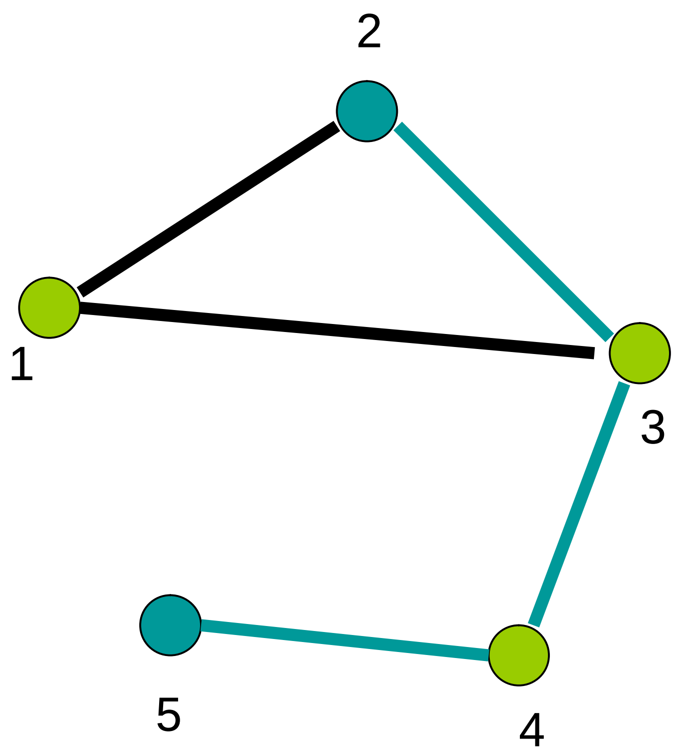
Shortest Paths

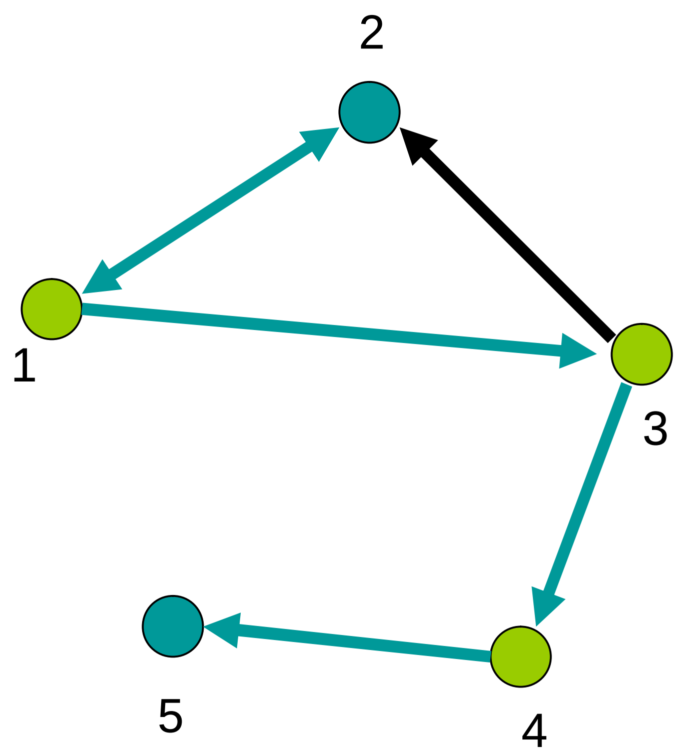
- Shortest Path from node i to node j
  - also known as BFS path, or geodesic path



Diameter

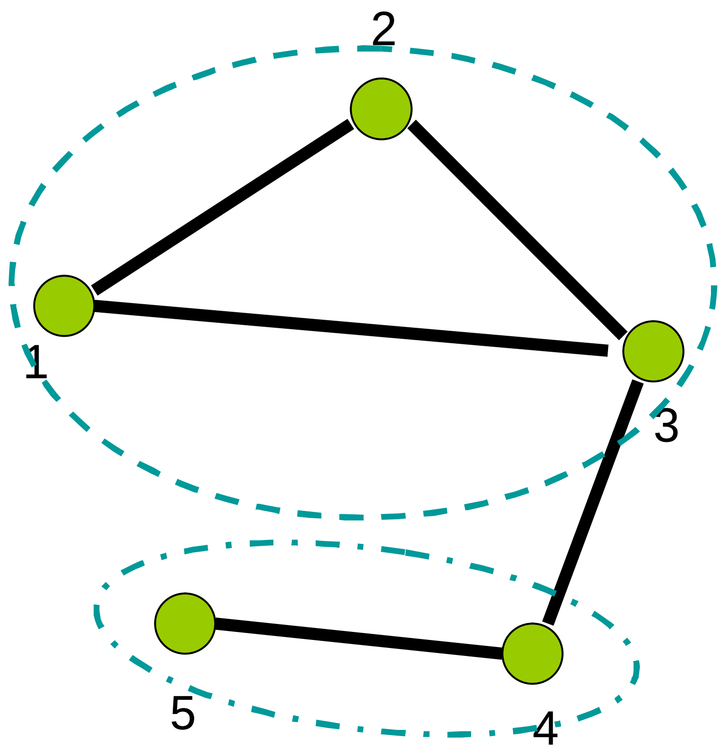
- § The longest shortest path in the graph





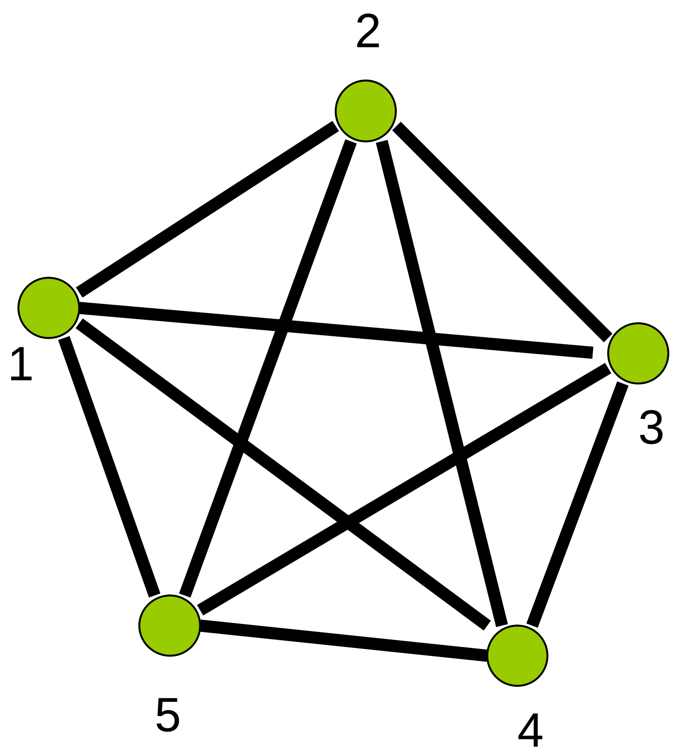
Undirected graph

- Connected graph: a graph where there every pair of nodes is connected
- Disconnected graph: a graph that is not connected
- Connected Components: subsets of vertices that are connected



Fully Connected Graph

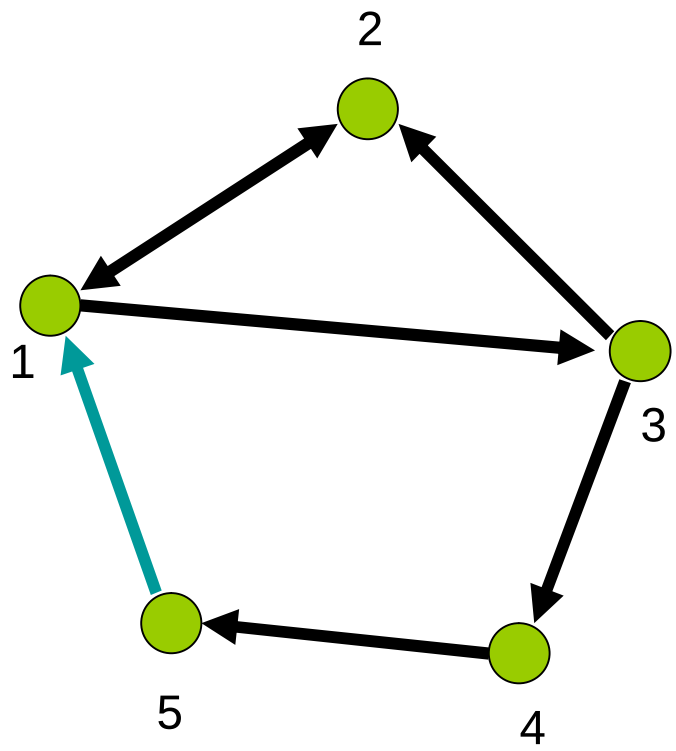
- Clique K_n
- A graph that has all possible n(n-1)/2 edges



Directed Graph

- Strongly connected graph: there exists a path from every i to every j

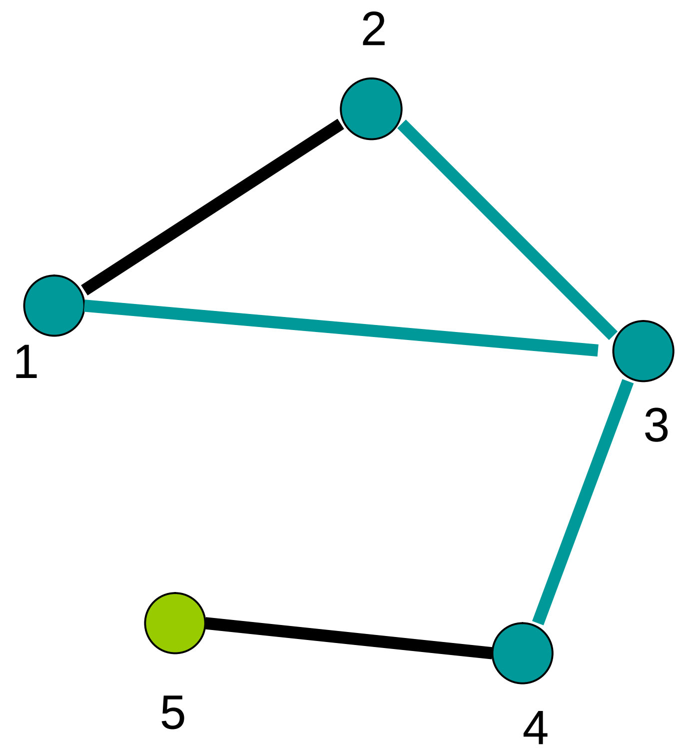
- Weakly connected graph: If edges are made to be undirected the graph is connected



Subgraphs

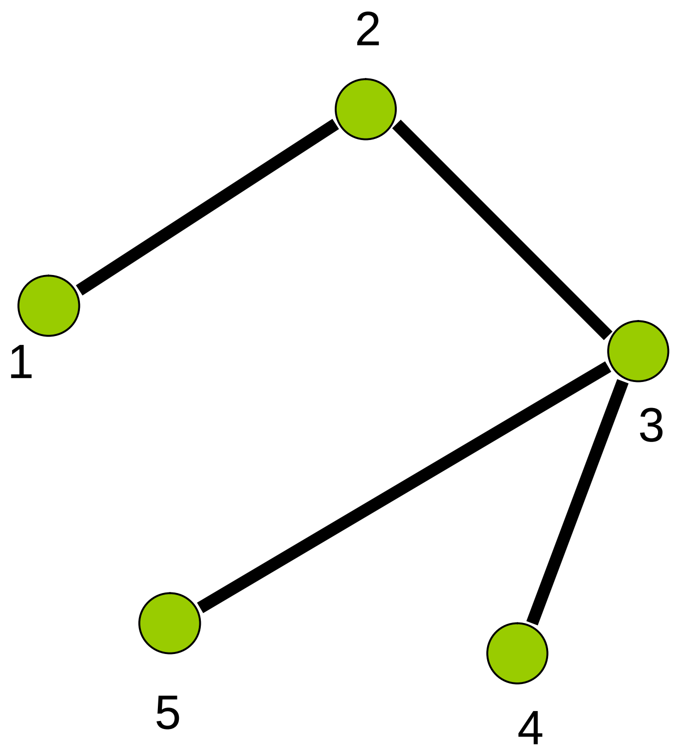
- Subgraph: Given V' $\subseteq$ V, and E' $\subseteq$ E, the graph G'=(V',E') is a subgraph of G.

- Induced subgraph: Given V' $\subseteq$ V, let E' $\subseteq$ E is the set of all edges between the nodes in V'. The graph G'=(V',E'), is an induced subgraph of G



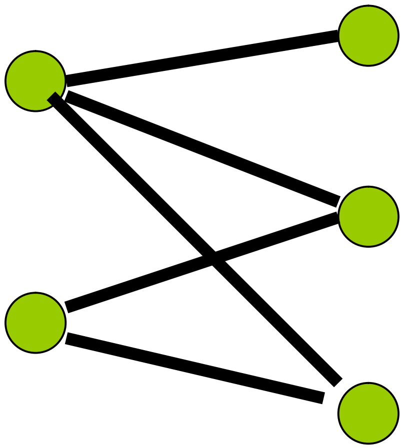
Trees

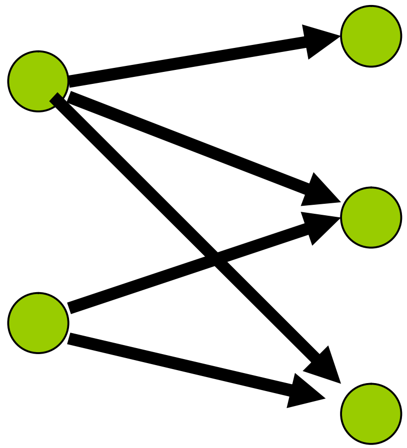
- Connected Undirected graphs without cycles



Bipartite graphs

- Graphs where the set V can be partitioned into two sets L and R, such that all edges are between nodes in L and R, and there is no edge within L or R

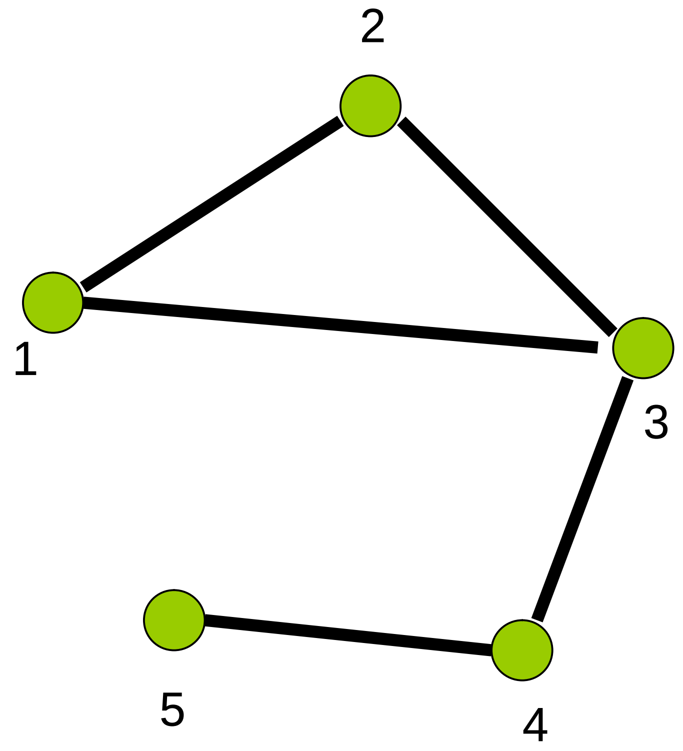




Linear Algebra

- Adjacency Matrix
  - symmetric matrix for undirected graphs

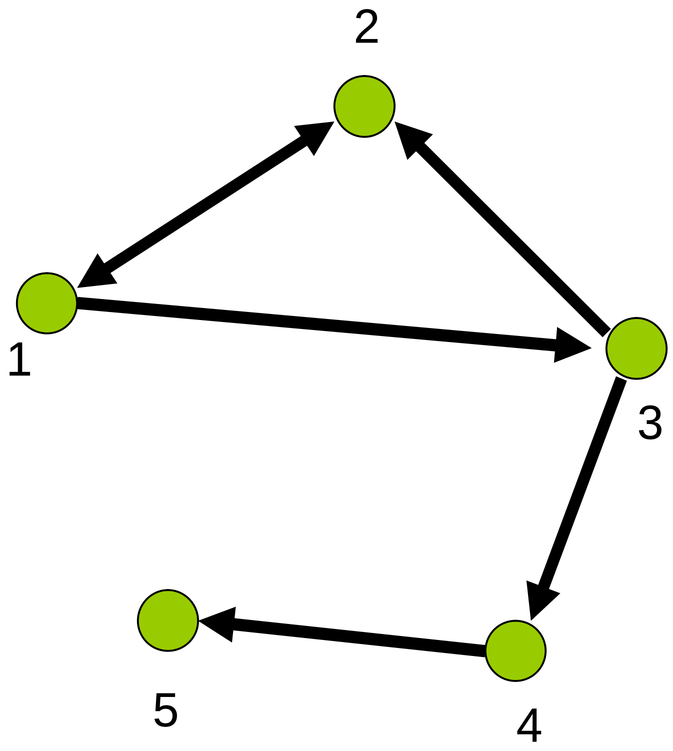
$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$



Linear Algebra

- Adjacency Matrix
  - unsymmetric matrix for undirected graphs

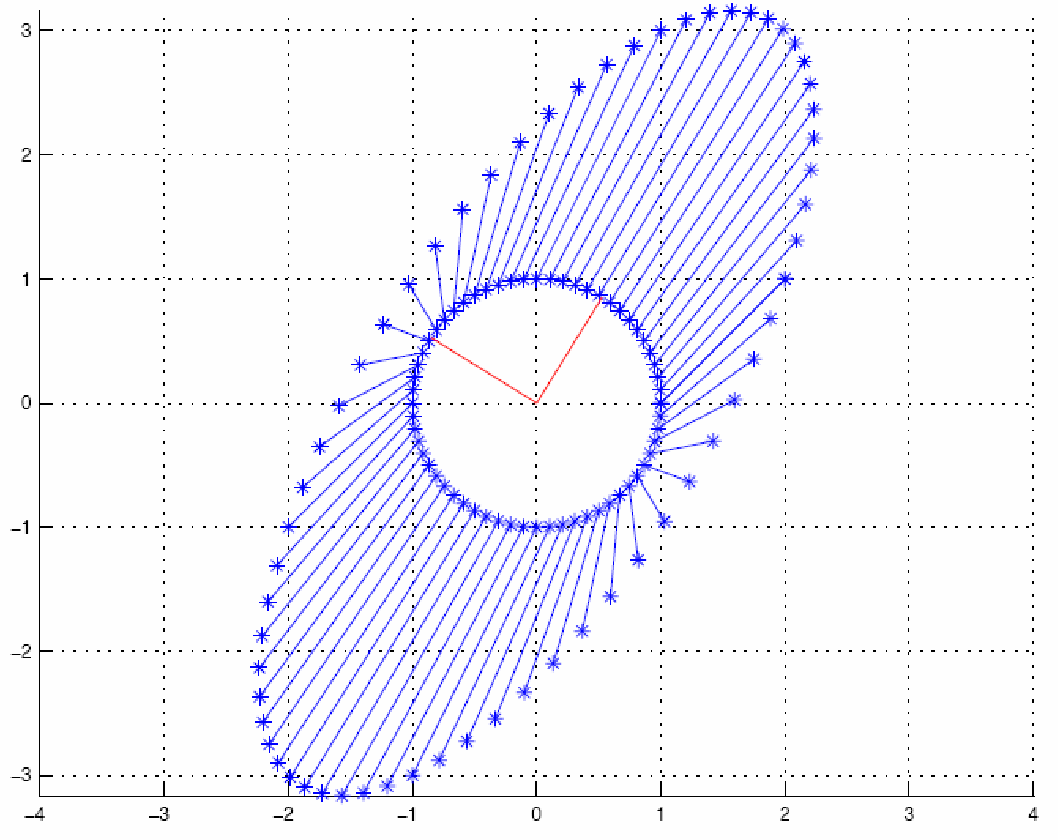
$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Eigenvalues and Eigenvectors

- The value λ is an eigenvalue of matrix A if there exists a non-zero vector x, such that $Ax=\lambda x$. Vector x is an eigenvector of matrix A
  - The largest eigenvalue is called the principal eigenvalue
  - The corresponding eigenvector is the principal eigenvector
  - Corresponds to the direction of maximum change

Eigenvalues



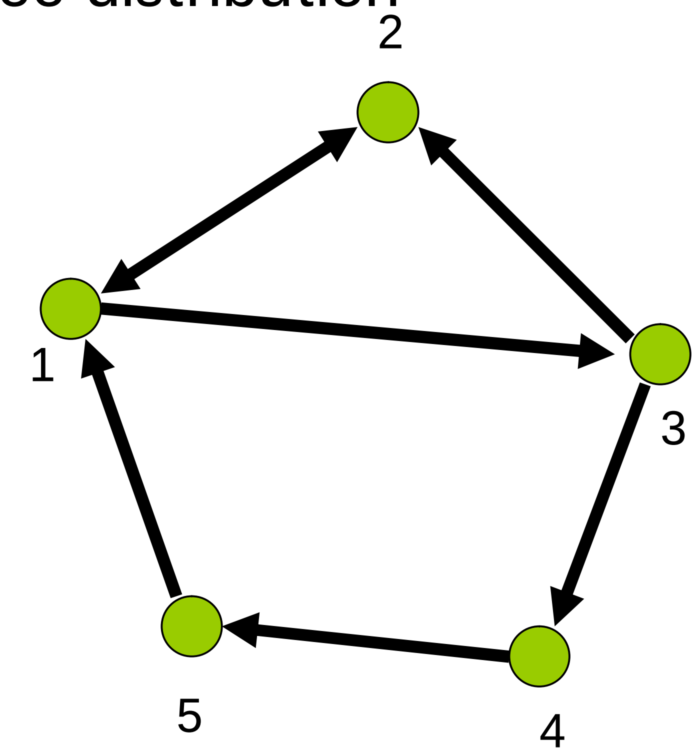
Random Walks

- Start from a node, and follow links uniformly at random.
- Stationary distribution: The fraction of times that you visit node i, as the number of steps of the random walk approaches infinity
  - if the graph is strongly connected, the stationary distribution converges to a unique vector.

Random Walks

- stationary distribution: principal left eigenvector of the normalized adjacency matrix
  - x = xP
  - for undirected graphs, the degree distribution

$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Probability Theory

- Probability Space: pair ‹Ω,P›
  - Ω: sample space
  - P: probability measure over subsets of Ω
- Random variable X: Ω→R
  - Probability mass function P[X=x]
- Expectation

$$E[X] = \sum_{x \in \Omega} x P[X = x]$$

Classes of random graphs

- A class of random graphs is defined as the pair ‹G_n,P› where G_n the set of all graphs of size n, and P a probability distribution over the set G_n
- Erdös-Renyi graphs: each edge appears with probability p
  - when $p=1/2$, we have a uniform distribution

Asymptotic Notation

- For two functions f(n) and g(n)
  - $f(n) = O(g(n))$ if there exist positive numbers c and N, such that $f(n) \le c\, g(n)$, for all $n \ge N$
  - $f(n) = \Omega(g(n))$ if there exist positive numbers c and N, such that $f(n) \ge c\, g(n)$, for all $n \ge N$
  - $f(n) = \Theta(g(n))$ if $f(n)=O(g(n))$ and $f(n)=\Omega(g(n))$
  - $f(n) = o(g(n))$ if $\lim f(n)/g(n) = 0$, as $n \to \infty$
  - $f(n) = \omega(g(n))$ if $\lim f(n)/g(n) = \infty$, as $n \to \infty$

P and NP

- P: the class of problems that can be solved in polynomial time
- NP: the class of problems that can be verified in polynomial time
- NP-hard: problems that are at least as hard as any problem in NP

Approximation Algorithms

- NP-optimization problem: Given an instance of the problem, find a solution that minimizes (or maximizes) an objective function.
- Algorithm A is a factor c approximation for a problem, if for every input x,

  $A(x) \leq c\ OPT(x)$ (minimization problem)

  $A(x) \geq c\ OPT(x)$ (maximization problem)

References

- M. E. J. Newman, The structure and function of complex networks, SIAM Reviews, 45(2): 167-256, 2003