Applied Matrix Analysis, MAA704

Christopher Engström

November 18, 2013

MAA704 Applied Matrix Analysis

Todays lecture:

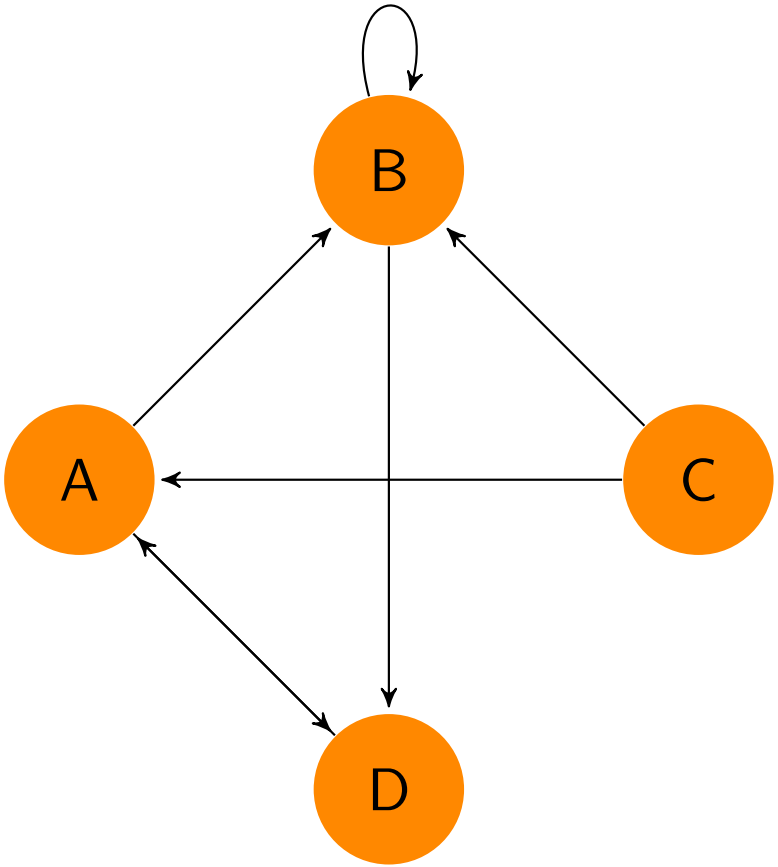
- Non-negative matrices and introduction to graph theory.
- Connectivity and Irreducibility.
- The Laplacian matrix of a graph

Non-negative matrices

- A non-negative matrix is a matrix A where all elements $a_{i,j} \geq 0$.
- A positive matrix is a matrix A where all elements $a_{i,j} > 0$.

Non-negative matrices in Graph-theory

A graph is a collection of vertices (nodes) and edges (links) such as the one below.



Non-negative matrices in Graph-theory

There are many types of graphs those we will work with are:

- **Simple** graphs, where we only allow a single edge from one vertice to another.
- **Undirected** graphs, where edges do not have a direction so we are only interested in if there is a edge between two vertices.
- **Directed graphs**, where edges do have a direction, a edge $A \to B$ does not necessary mean there is a edge $B \to A$.
- **Weighted graph**, where we assign (positive) weights as scalars to every edge.

Non-negative matrices in Graph-theory

How can we represent a (simple) graph using a matrix?

- Multiple ways such as the adjacency matrix, distance matrix, incidence matrix, Laplacian matrix, etc.
- Which graph type and which matrix representation you use depend on your application.

Application, road network and connectivity

We consider a road network between different cities where we want to know if we can take the from one city to another, or if we need to go by for example air/boat.

- We represent the road network using a undirected graph by letting the vertices represent the cities, and a edge between two cities means there is a road between the two cities.
- If there is a edge between our two cities A, B we know there is a road and we can take the car. However if there isn't we might still be able to take the car if for example there is a road between A and another city C and also a road between C and B.
- To solve this problem we define connectivity for graphs, and irreducibility for matrices.

Connectivity

- Two vertices is said to be **connected** if by traversing the edges there exists a path from both of the nodes to the other.
- A graph is said to be connected if it is **connected** for all pair of nodes in the undirected graph.
- A directed graph is said to **strongly connected** if it is **connected** for all pair of nodes in the directed graph..

Connectivity: connected components

A **connected component** in a undirected graph is a maximal part of the graph where all nodes are connected with eachother.

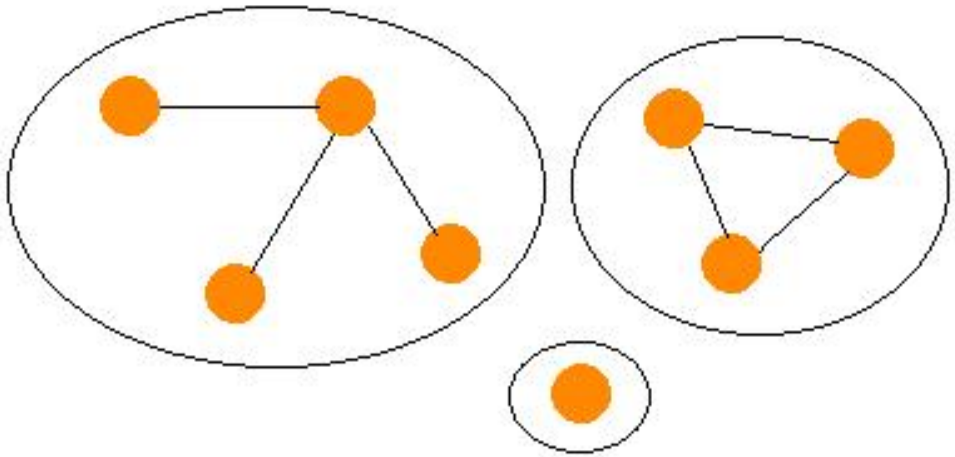


Figure: A undirected graph with 3 connected components

Connectedness: strongly connected components

A **strongly connected component** is a part of the graph which is strongly connected.

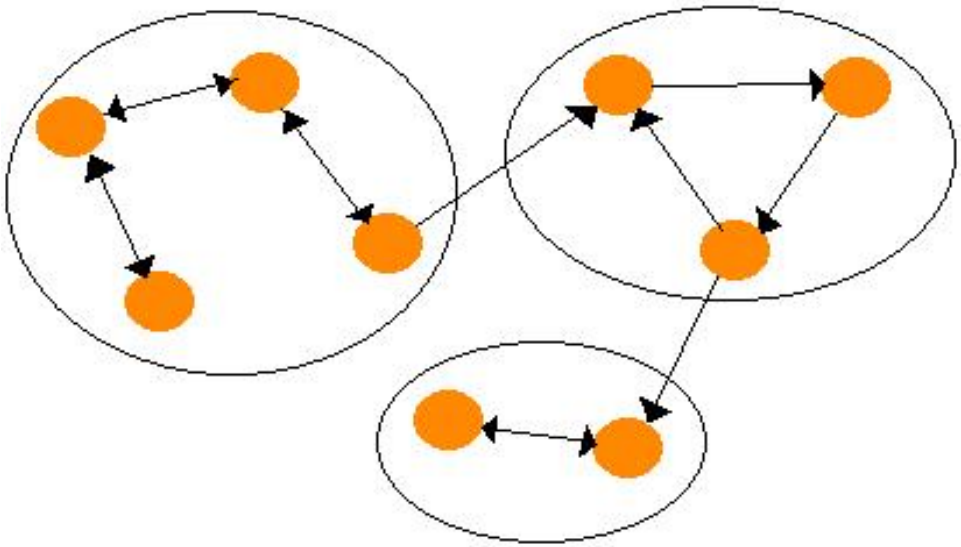


Figure: A directed graph with 3 strongly connected components

Application, road network and adjacency matrix

So if our two cities belong to the same connected component, there is a path between them and we can take the car.

- But how do we find the connected components? Especially if we have a large graph it might even be hard to visualize the graph.
- To solve the problem we use a matrix representation called the adjacency matrix.

Application, road network and adjacency matrix

Definition

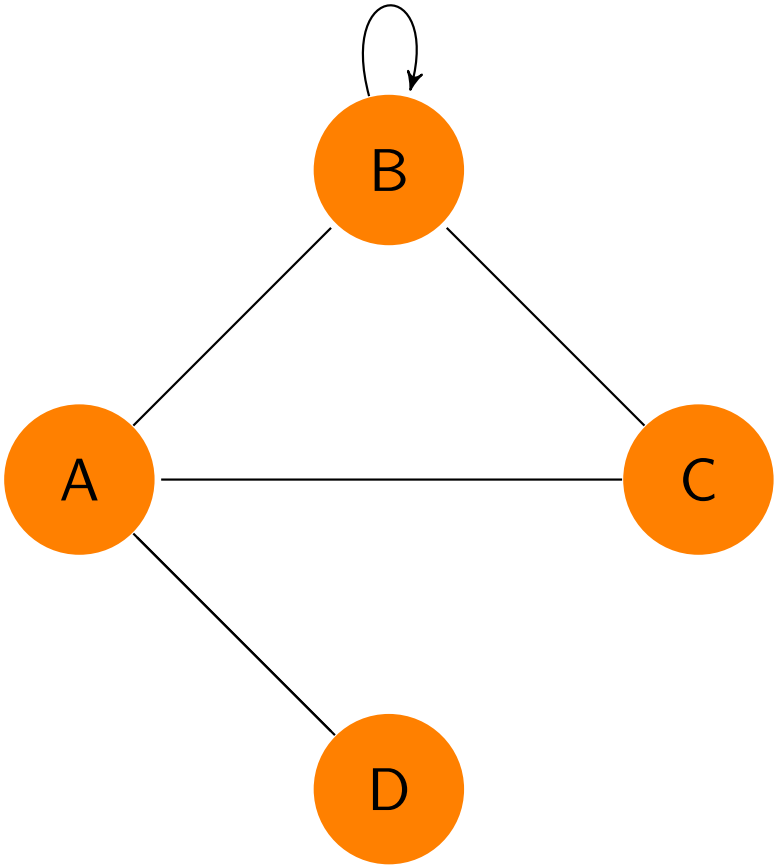
The **adjacency matrix** A of a graph G with n vertices is a square $n \times n$ matrix with elements $a_{i,j}$ such that:

$$a_{i,j} = \begin{cases} 1, & \text{if there is a link from vertice } i \text{ to vertice } j \\ 0, & \text{otherwise} \end{cases}$$

If the graph is undirected we consider every edge between two vertices as linking in both directions.

Non-negative matrices in graph theory

Example of an undirected graph and corresponding adjacency matrix (vertices ordered A,B,C,D).



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

Application, road network and adjacency matrix

We note that the adjacency matrix is not unique in itself, we need to choose in what order we put our vertices in the matrix!

- Sometimes changing the order of the vertices (essentially re-labeling the graph) can make certain structures more obvious.
- For example we could group vertices in the same connected component together.

Adjacency matrix, other applications

A undirected or directed graph and it's adjacency matrix could also be used to represent other things such as:

- Links between homepages, and PageRank.
- Electrical or water flow networks.
- Linguistic relations between words or phrases.

Application, road network and adjacency matrix

Returning to our road network example, while a 1 at element a_{ij} means there is a road between those two cities, we still don't know how to find the connected components using the adjacency matrix.

- We take a look at the powers A^k of the adjacency matrix.
- A non-zero element of A^1 means there is a path of "length" 1 between the two vertices. (length as in number of edges)

Application, road network and adjacency matrix

We look at A^2 and when a single element $a_{ij}^2 > 0$.

- For $a_{i,j}^2 > 0$ we need that at least one of $a_{i,k}a_{k,j} > 0, k = 1, 2, \ldots, n$.
- $a_{i,k} = 1$ if there is a edge between i, k and $a_{k,j} = 1$ if there is a edge between k, j.
- So for $a_{i,j}^2 > 0$ we need at least one path of exactly length 2 between i, j.
- It's easy to see using for example induction that the same is true for A^k, $k > 0$ as well.

Application, road network and adjacency matrix

While we can compute all of $A, A^2, A^3, \ldots, A^k$ for some large k and see if any $a_{i,j} > 0$ for any k, there is a better way.

- We notice that if we let every vertice link to itself, then if there is a path from one vertice to another of length k, then there is a path of length $k+1$ as well simply by looping in the last vertice once.
- Additionally the longest possible shortest path is $n-1$, since we after traveling to $n-1$ new vertices we need to return to one we have already visited.
- Looking at the elements of $(I+A)^{n-1}$ we can thus see if there is at least one path between any two vertices.

Irreducible non-negative matrices.

Theorem

A non-negative square $n \times n$ matrix A is said to be **Irreducible** *if any of the following equivalent properties is true:*

1. *The graph corresponding to A is strongly connected.*
2. *For every pair i, j there exists a natural number m such that $\mathrm{A}^m_{i,j} > 0$.*
3. *$(\mathrm{I} + \mathrm{A})^{n-1}$ is a positive matrix.*
4. *A cannot be conjugated into a block upper triangular matrix by a permutation matrix P.*

Irreducible non-negative matrices.

Irreducible non-negative matrices and their properties are used extensively in for example:

- Perron-Frobenius theory.
- Stochastic processes.
- Markov chains and Markov chain monte carlo.

Vertex and edge connectivity

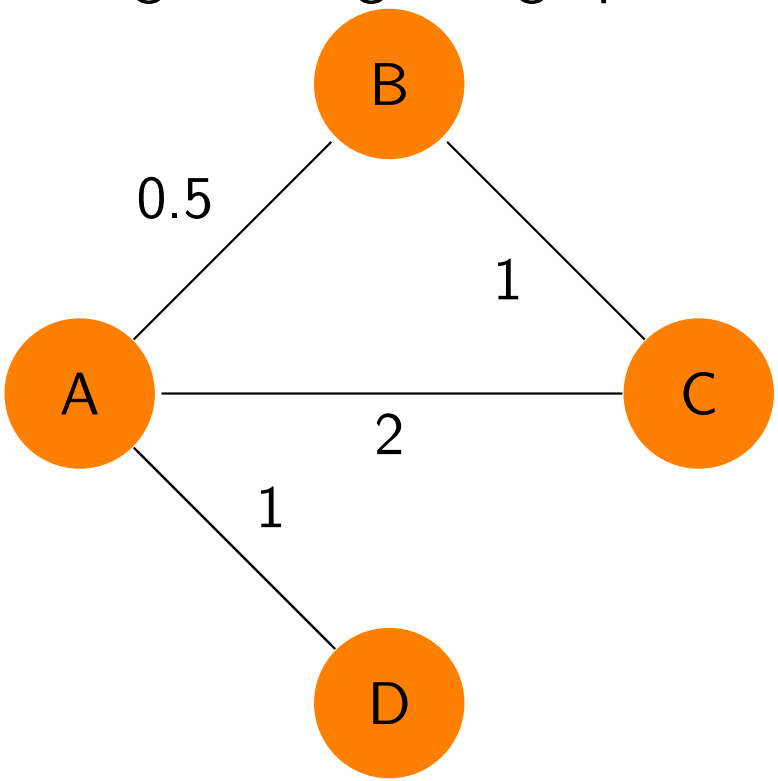
Even if the graph is **strongly connected** it might not be very well connected or it might be interesting to see how easy it is to divide the graph in two disjoint parts.

- The **vertex connectivity** of a graph is the minimum number of vertices whose removal makes the graph not **strongly connected** anymore.
- The **edge connectivity** of a graph is the minimum number of edges whose removal makes the graph not **strongly connected** anymore.
- Finding he edge connectivity is related to the max-flow min-cut problem.

Distance matrix

While good to know whether there exist a path between two vertices, the adjacency matrix says nothing of the length of those paths.

- To do this we introduce (positive) weights on the edges representing the distance between two cities.
- This gives weighted graph such as:



Distance matrix

To represent this weighted graph we only need to make a short modification of the adjacency matrix:

Definition

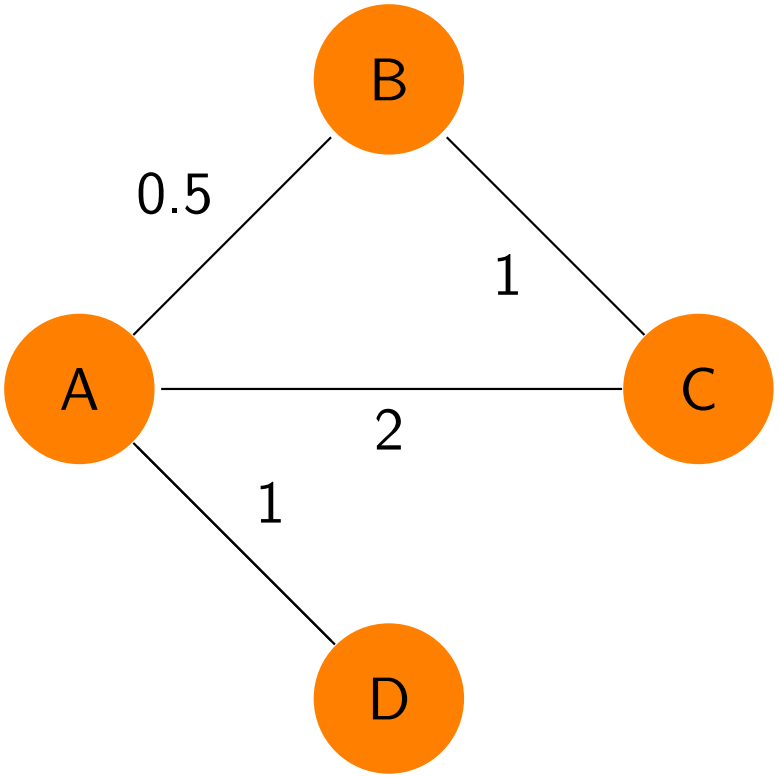
The **distance matrix** D of a graph G with n vertices and weighted edges E is a square $n \times n$ matrix with elements $d_{i,j}$ such that:

$$d_{i,j} = \begin{cases} E_{i,j}, & \text{if there is a link from vertice } i \text{ to vertice } j \\ 0, & \text{otherwise} \end{cases}$$

Where $E_{i,j}$ is the weight of the edge from vertice i to vertice j. As with the adjacency matrix, if the graph is undirected we consider every edge between two vertices as linking in both directions.

Distance matrix

Here is an example of the previous graph and corresponding distance matrix.



$$\begin{bmatrix} 0 & 0.5 & 2 & 1 \\ 0.5 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Weighted graphs and applications

A weighted graph and it's distance matrix can for example be used to represent:

- Distance between vertices physical or some other defined distance, or time needed to travel from one vertice to another.
- Electrical networks, where the weights represent the resistance over the wires.
- Water flow network with capacities represented by the weights.
- Markov processes and random walks where the weights represent a probability to move from one state to another.

Distance matrix and applications

Using the distance matrix we can do the same analysis as we did for the adjacency matrix to for example see if the matrix is irreducible.
However the distance matrix can also be used for such as:

- Finding the shortest path between two or more vertices.
- Finding stationary distributions, hitting times, hitting probabilities, and much more for Markov chains.
- Calculating the potential of electrical networks.

Application: shortest path

- A common algorithm to find the shortest distance between one source node and one or all other nodes is Djikstra's algorithm.
- It works by assigning a distance from the initial node to every other node and then travel through the graph by always choosing the next node among the unvisited nodes that minimizes the distance.
- At every node it updates the distance to every node linked to by the current node.

Application: Djikstra's algorithm

More exact the method works through the following steps:

1. Assing distance zero to initial node and infinity to all other nodes.
2. Mark all nodes as unvisited and set the initial node as current node.
3. For the current node, calculate the distance to all it's unvisited neighbors as the current distance plus the distance to the neighbors. Update these nodes distance if the newly found distance is lower.
4. Mark current node as visited.
5. If the smallest distance in among the unvisited nodes is infinity: stop! Otherwise set the unvisited node with the smallest distance as current node and go to step 3.

Application: Djikstra's algorithm

Think about the following and if or how it effects the usefulness of Djikstra's algorithm.

- What happens if there are negative weight's in the graph.
- What if the graph is directed? Should a transportation network be considered directed?
- What if the graph is irreducible? What if it isn't?

Laplacian matrix and degree matrix of a graph

The **Laplacian matrix** of a graph is very useful for finding a number of usefull properties of a graph. To find the laplacian matrix we first need to know the degree matrix:

- The **degree matrix** of a undirected graph is a diagonal matrix where the elements on the diagonal is the number of edges connected with that node.
- In other words given a graph G with n nodes the degree matrix D is a $n \times n$ matrix with elements:

$$d_{i,j} := \begin{cases} \deg(i), & i = j \\ 0, & i \neq j \end{cases}$$

  Where $deg(i)$ is the number of edges connected with vertice v_i.
- If there are loops present (nodes linking to itself) we count those twice for the degree.

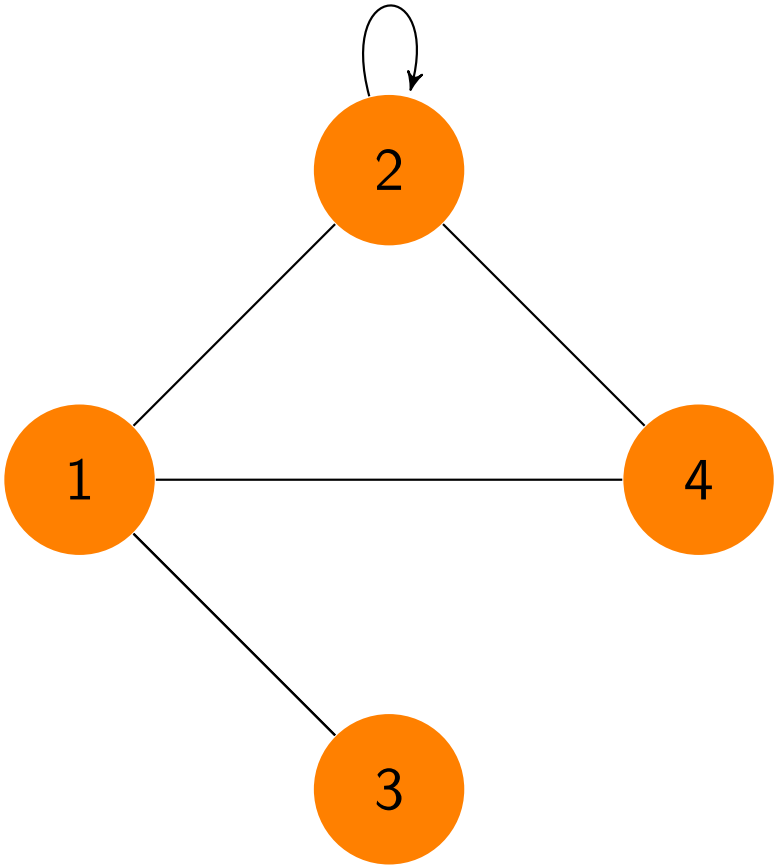
Laplacian matrix and degree matrix of a graph

If the graph is directed we make the following modifications:

- We use either in-degree (number of links from other nodes to target node) or out-degree (number of links from node to other nodes).
- Which one we use depend on the application.
- We do not count loops twice if the graph is directed.

The degree matrix: undirected graph

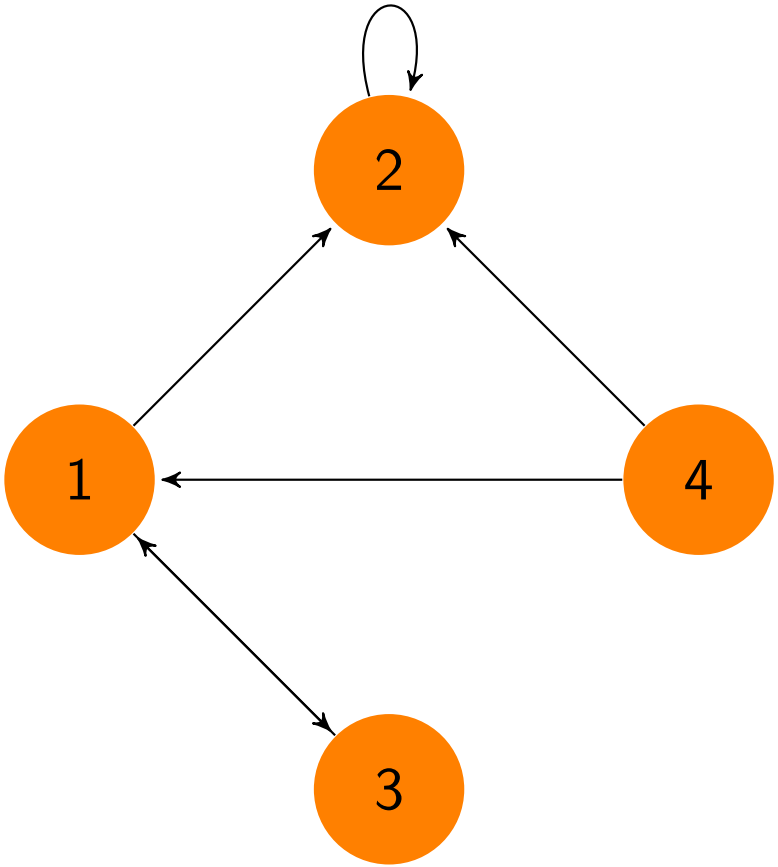
Degree matrix of an undirected graph.



$$D = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

The degree matrix: directed graph

Degree matrix using in-degree.



$$D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Laplacian matrix

The **Laplacian matrix** L of a graph with no self loops is defined as:

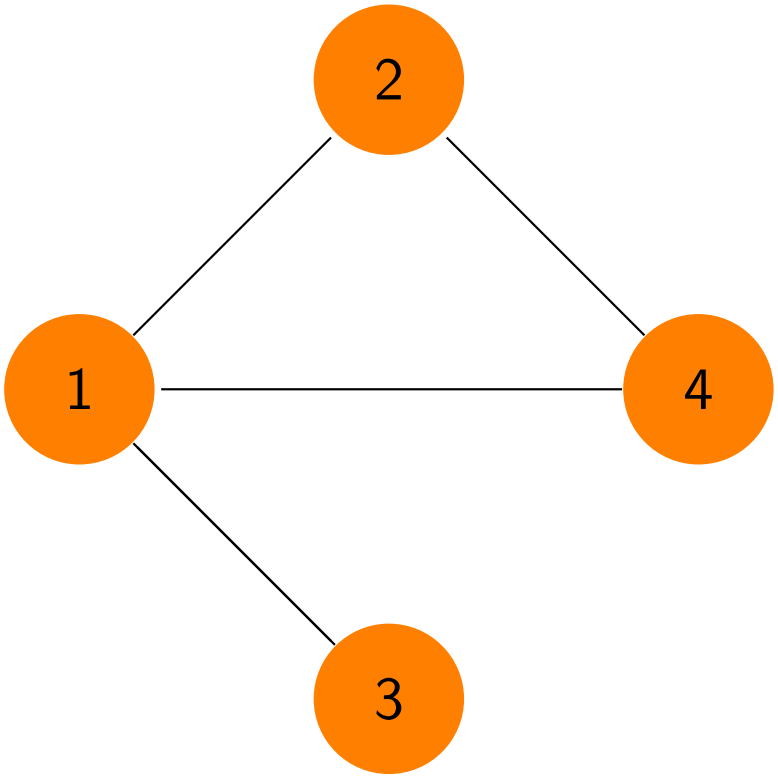
$$\mathrm{L} := \mathrm{D} - \mathrm{A}$$

where D is the **degree matrix** and A is the **adjacency matrix**. This gives elements $l_{i,j} \in \mathrm{L}$:

$$l_{i,j} := \begin{cases} \deg(i), & i = j \\ -1, & i \neq j, v_i \text{ links to } v_j \\ 0, & \text{otherwise} \end{cases}$$

Where $deg(i)$ is the number of edges connected with vertice v_i.

Laplacian matrix



$$L = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ -1 & -1 & 0 & 2 \end{bmatrix}$$

Laplacian matrix

Some properties of the Laplacian matrix L and it's eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.

- L is positive semidefinite ($\mathbf{x}^T \mathrm{L}\mathbf{x} \geq 0$).
- $\lambda_1 = 0$ and the number of connected components is it's algebraic multiplicity.
- The smallest non zero eigenvalue is called the spectral gap.
- The second smallest eigenvalue of L is the algebraic connectivity of the graph.

Laplacian matrix: spectral gap

The spectral gap can be used to for example:

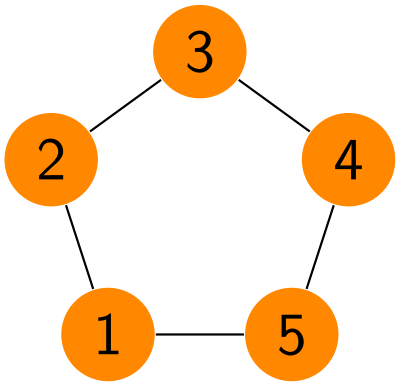
- Using Kirchhoff's theorem: Find the number of spanning trees (group of edges without any periods which connect all vertices).
- Using Cheeger's inequality: Estimate whether the network have "bottlenecks" or is well connected.

Laplacian matrix: algebraic connectivity

- The algebraic connectivity is greater than 0 only if the graph is Irreducible. It is bounded above by the vertex connectivity.
- The algebraic connectivity is a value which gives an indication on how connected the graph is, which helps in evaluating the robustness or how easy the network synchronize.
- The eigenvector called the Fiedler-vector to this eigenvalue have uses in partitioning graphs.

Application: Robustness of a network.

We look at the network described by the graph and it's Laplacian matrix below:



$$L = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}$$

Application: Robustness of a network.

We look at the connectedness of the graph:

- It's easy to see that both the vertex connectivity and edge connectivity are 2.
- This stays true even if we increase the number of vertices in the graph.
- This means that either 2 vertices or 2 edges would need to fail for the network to be divided in two parts.

Application: Robustness of a network.

- If we calculate the eigenvalues and look at the second smallest eigenvalue $\lambda_2 = 1.38$.
- The algebraic connectivity is therefor 1.38.
- However if we increase the number of nodes the algebraic connectivity decreases.
- At 6 nodes we have $\lambda_2 = 1$ and at 10 nodes we have $\lambda_2 = 0.38$.

Next lecture

Next lecture we will look at:

- Perron-Frobenius and properties of non-negative matrices.
- Markov chains, stochastic matrices and relation to graph theory.
- Using Perron-Frobenius theory for stochastic matrices and applications of stochastic matrices.