Graphs and Linear Algebra

- Many interesting properties of a graph $G = (V, E)$ can be studied using *linear algebra*
- Example: weighted graphs corresponding to markov models, stationary distribution is related to the eigenvector of the adjacency matrix with eigenvalue 1
- Application: What is the limiting behavior of a random walker on a given graph

Adjacency, Degree, and Laplacian

- Matrices associated to a graph G
- Adjacency Matrix A: indexed by nodes, 1 if there is edge between (u, v) 0 otherwise
- Degree Matrix D: Diagonal matrix with the node *degree* placed at corresponding entry $D_{uu} =$ degree of u
- The difference $D - A$ is called the graph *Laplacian*

Connected Components of Graphs

- Connected Component: subset of the nodes where any two are connected by a path in the graph
- Topological invariant of a graph
- Breadth first or Depth first search beginning at some node will return the connected component that contains the start node

Relationship of graph laplacian and connected components

- Eigenvalue analysis of the the laplacian L reveals interesting properties about the related graph G
- Eigenvalues of L are all real
- L is positive semidefinite (all eigenvalues are non-negative)
- The number of times 0 appears as an eigenvalue of L is equal to the number of connected components in G

Relationship to the Continuous Version of the Laplacian

- In physics (partial differential equations) the laplacian often appears in the equations describing natural systems
- Heat Equation, Wave Equation
- In general, what kind of phenomenon does the *heat equation* describe

Laplacian in networkx

- networkx provides various linear algebra functions for graphs

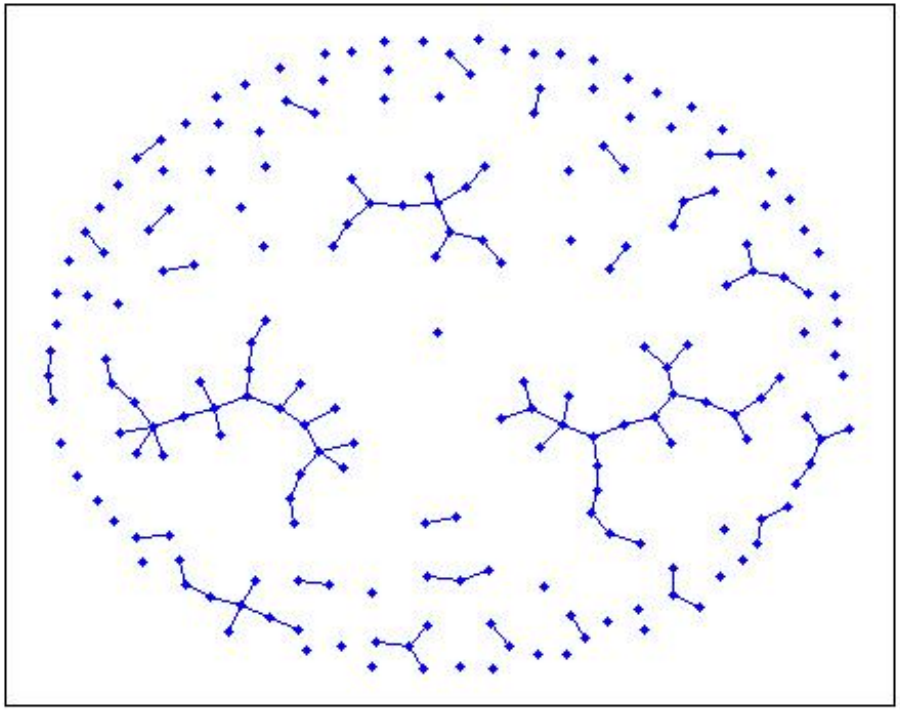
```
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np

<... define some Graph objects ...>

a = nx.adjacency_matrix(L)
print(a)

l = nx.laplacian_matrix(K)
l = l.todense()
w, v = np.linalg.eig(l)
print(w)
```

Erdos-Renyi model



- Erdos-Renyi model constructs a random graph where each edge exists independently with probability p
- Reasonable sampling of graphs in general

In-Class Exercise

- Compute the spectrum of some graphs in using the `networkx` package in python
- In particular write a script that generates a ER graphs with different parameters (probability of having an edge)
- Compute the graph laplacian, perform an eigenanalysis
- Verify that the eigenvalues of L behave as expected