MATH 168 - Networks : HW 2

Dhruv Chakraborty
Prof. Mason Porter
Fall 2020

Networks: Problem Set 2

2) Newman 6.3

(a) $A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Note: directed network so $A_{ij} = 1$ if edge from j to i

(b) $B = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix}$

(c) $P = B^T B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix}$

$\therefore P = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 \\ 1 & 2 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$

3) Newman 6.6

Note that by def, a star graph's adjacency matrix must appear as follows:

$$A = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & & & \\ \vdots & & \mathbf{0} & \\ 1 & & & \end{pmatrix} \Bigg\} n$$

$$\underbrace{\qquad\qquad}_{n}$$

where in our representation the first row depicts the central node.

Let λ be the largest eigenvalue of A and let $\vec{v}$ be an associated eigenvector.

Then, we have by def that $A\vec{v} = \lambda\vec{v}$

i.e. $$\begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & & & \\ \vdots & & \mathbf{0} & \\ 1 & & & \end{pmatrix} \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \lambda \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} v_2 + v_3 + \cdots + v_n \\ v_1 \\ \vdots \\ v_1 \end{pmatrix} = \lambda \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

But this means $\lambda v_1 = v_2 + v_3 + \cdots + v_{n-1} + v_n$ (✡)

and $v_1 = \lambda v_2 = \lambda v_3 = \dots = \lambda v_n$

$$\Downarrow$$

$$v_2 = v_3 = \dots = v_n = v_1/\lambda$$

Substituting in ✡, $\lambda v_1 = \underbrace{\frac{v_1}{\lambda} + \frac{v_1}{\lambda} + \dots + \frac{v_1}{\lambda}}_{n-1 \text{ times}}$

$$\Rightarrow \lambda^2 v_1 = (n-1) v_1$$

$$\Rightarrow \underline{\lambda = \sqrt{n-1}} \text{ for largest eigenvalue}$$

4) Newman 6.12

(a) We first note that the sum S of the weights of all paths from s to t $= \sum_{r=0}^{\infty} N_{st}^r \alpha^r$ where N_{st}^r is the number of paths from s to t in the network. But we also know that

$$A_{st}^r = N_{st}^r \quad \therefore S = \sum_{r=0}^{\infty} \alpha^r A_{st}^r$$

Let $S = \sum_{r=0}^{\infty} x_{st}^r$ where $x = \alpha A$

But recall the Taylor series expansion

$$\frac{1}{1-x} = \sum_{r=0}^{\infty} x^r \quad \text{so we get}$$

$$S = \sum_{r=0}^{\infty} (\alpha A)^r_{st} = \left(\frac{1}{1-\alpha A}\right)_{st} = (1-\alpha A)^{-1}_{st} = Z_{st} \quad \square$$

(b) we know that for $\sum_{r=0}^{\infty} x^r$ to converge to $\frac{1}{1-x}$ we must have $-1 < x < 1$. Noting that our networks paths are weighted α^r for r hops, A has only 0s and 1s. So for $x = \alpha A$ we must have $|\alpha| < 1$ for the sum to converge.

c) using the chain rule we know

$$\frac{\partial \log Z_{st}}{\partial \log \alpha} = \frac{\frac{\partial \log Z_{st}}{\partial Z_{st}} \frac{\partial Z_{st}}{\partial \alpha}}{\frac{\partial \log \alpha}{\partial \alpha}}$$

$$(\star) \quad = \frac{\frac{1}{Z_{st}} \frac{\partial Z_{st}}{\partial \alpha}}{\frac{1}{\alpha}} = \frac{\alpha}{Z_{st}} \frac{\partial Z_{st}}{\partial \alpha}$$

Since ℓ_{st} is the length of the shortest path, using our sum above we also know

$$\sum_{r=0}^{\ell_{st}-1} \alpha^r A^r_{st} = 0$$

Using this knowledge then our new sum $S =$

$$Z_{st} = \sum_{r=l_{st}}^{\infty} \alpha^r A_{st}^r = \sum_{r=0}^{\infty} \alpha^{r+l_{st}} A_{st}^{r+l_{st}}$$

Taking the derivative of both sides wrt α:

$$\frac{\partial Z_{st}}{\partial \alpha} = \frac{\partial}{\partial \alpha}\left(\alpha^{l_{st}} \sum_{r=0}^{\infty} \alpha^r A_{st}^{r+l_{st}}\right)$$

$$= \left(\sum_{r=0}^{\infty} \alpha^r A_{st}^{r+l_{st}}\right) l_{st} \alpha^{l_{st}-1} + \alpha^{l_{st}} \sum_{r=1}^{l_{st}} r A_{st}^{r+l_{st}} \alpha^{r-1}$$

Substituting this in ✡, we get:

$$\frac{\partial \log Z_{st}}{\partial \log \alpha} = \frac{\alpha}{Z_{st}}\left[l_{st}\alpha^{l_{st}-1} \sum_{r=0}^{\infty} \alpha^r A_{st}^{r+l_{st}} + \alpha^{l_{st}} \sum_{r=1}^{l_{st}} \alpha^{r-1} r A_{st}^{r+l_{st}}\right]$$

$$\Rightarrow \frac{\partial \log Z_{st}}{\partial \log \alpha} = \frac{\alpha}{\sum_{r=0}^{\infty} \alpha^r A_{st}^r}\left[l_{st}\alpha^{l_{st}-1} \sum_{r=0}^{\infty} \alpha^r A_{st}^{r+l_{st}} + \alpha^{l_{st}} \sum_{r=1}^{l_{st}} r\alpha^{r-1} A_{st}^{r+l_{st}}\right]$$

$$= l_{st} + \frac{\alpha^{l_{st}} \sum_{r=1}^{l_{st}} r\alpha^{r-1} A_{st}^{r+l_{st}}}{\sum_{r=0}^{\infty} \alpha^r A_{st}^r}$$

as $\alpha \to 0$, numerator $\to 0$

$\therefore$ this term vanishes

$$\therefore \lim_{\alpha \to 0} \frac{\partial \log Z_{st}}{\partial \log \alpha} = l_{st}$$

5) Random Graphs

(a) $P(|E| = m) = \binom{\binom{N}{2}}{m} p^m (1-p)^{\binom{N}{2}-m}$

Then, $\langle m \rangle = \sum_{m=0}^{\binom{N}{2}} m P(|E| = m) = \binom{N}{2} p$

which we can also see since there are $\binom{n}{2}$ possible edges, each appearing with probability p (using linearity of expectation).

(b) Note that any node i in G has an edge with each of the other $n-1$ nodes with probability p.

So degree$(i) \sim$ Binomial$(n-1, p)$

i.e. $P(\text{degree}(i) = d) = \binom{n-1}{d} p^d (1-p)^{n-1-d}$

$\therefore$ Mean degree of an ER graph

$$E(\text{degree}(i)) = (n-1)p$$

(c) For large n and small p we can use the Poisson approximation to the binomial distribution i.e. for small and positive p as $n \to \infty$

Binomial(n,p) may be approx. by Poisson(np)

$\therefore P_k \sim \text{Poisson}(c = (N-1)p)$

$= e^{-c} \frac{c^k}{k!}$ where k is the degree

(d) Note: my Python code, a visualization of a $G \sim G(10, 0.5)$ graph and some density plots (for mean degree and mean num of edges in sample) are attached below.

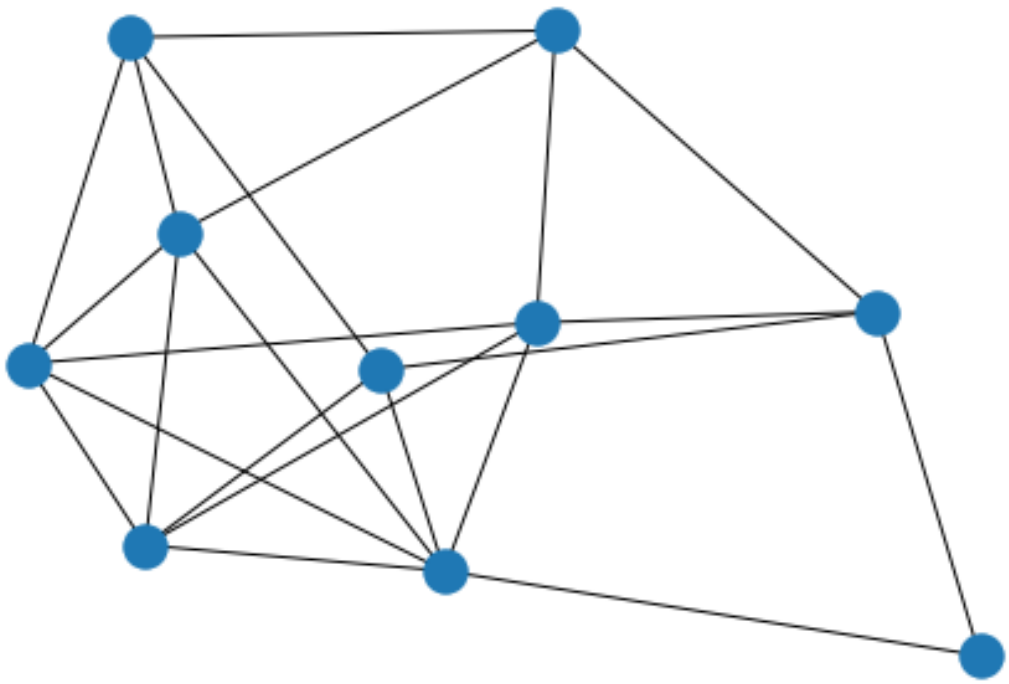
- I first sample 1000 $G(100, 0.5)$ graphs resulting in mean degree 49.50 (very close to expected value of 50) and mean num edges 2475.10 (very close to $E(|E|) = 2475$)
- For more examples I also sampled 100 $G(100, 0.5)$ graphs, 100 $G(1000, 0.5)$ graphs and 10000 $G(100, 0.5)$ graphs. All these had mean degree and num edges very close to the expected value. As we take the sample mean over a larger number of ER graphs we expect them to get even closer to the expected values.

Math 168 HW 2

October 19, 2020

[9]:
```
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[3]:
```
G = nx.erdos_renyi_graph(10, 0.5)
nx.draw(G)
plt.show()
```



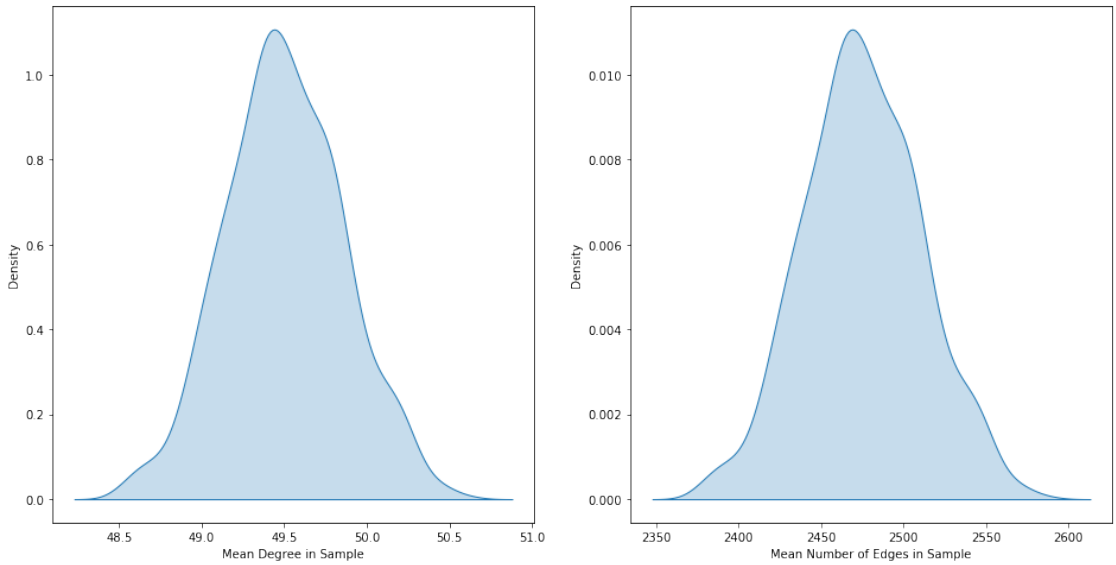
[32]:
```
def sample_ER_graphs(num_graphs, n, p=0.5):
    degrees, num_edges = [], []
    for _ in range(num_graphs):
```

```
        G = nx.erdos_renyi_graph(n, p)
        means = (np.mean(nx.degree(G)), nx.number_of_edges(G))
        degrees.append(means[0])
        num_edges.append(means[1])

    fig, axes = plt.subplots(1,2, figsize=(16,8))
    sns.kdeplot(ax=axes[0], x=degrees, fill=True)
    axes[0].set(xlabel="Mean Degree in Sample")
    sns.kdeplot(ax=axes[1], x=num_edges, fill=True)
    axes[1].set(xlabel="Mean Number of Edges in Sample")
    plt.show()

    return(degrees, num_edges)
```

[33]:
```
degrees, num_edges = sample_ER_graphs(1000, 100)
```

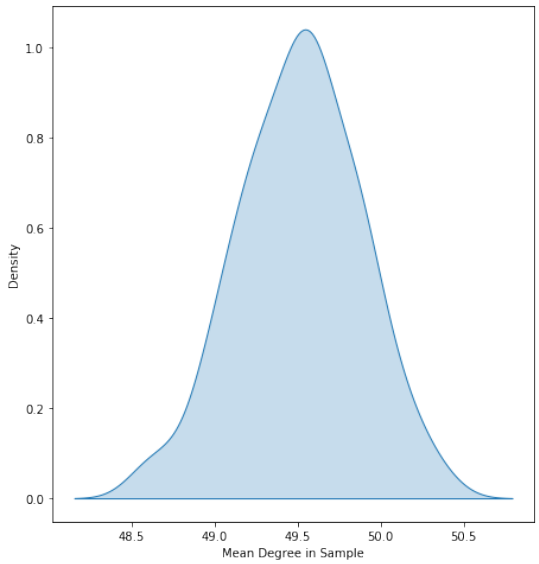


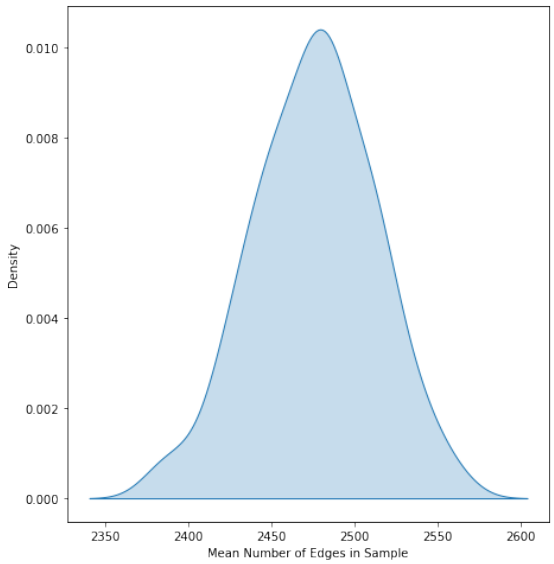
[36]:
```
display(np.mean(degrees), np.mean(num_edges))
```

```
49.50103
```

```
2475.103
```

[37]:
```
degrees, num_edges = sample_ER_graphs(100, 100)
```



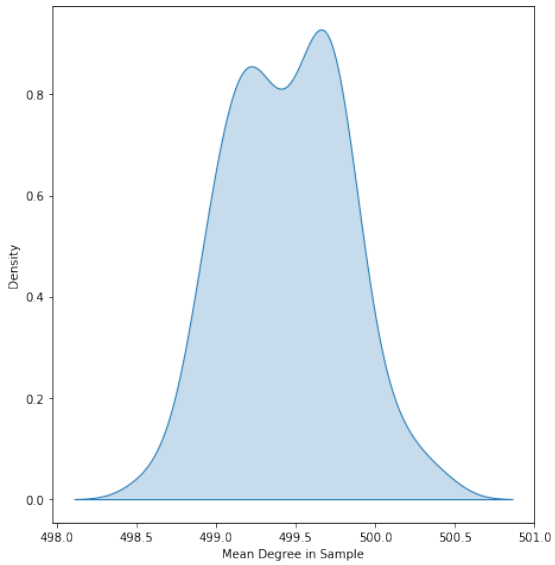


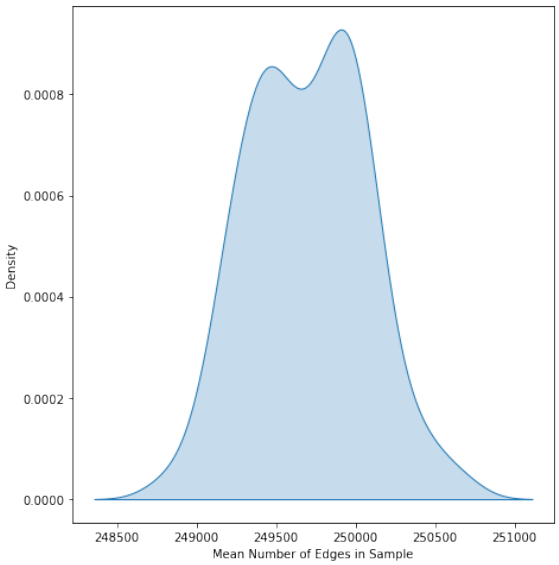
[38]:
```
display(np.mean(degrees), np.mean(num_edges))
```

49.51079999999999

2476.08

[39]:
```
degrees, num_edges = sample_ER_graphs(100, 1000)
```



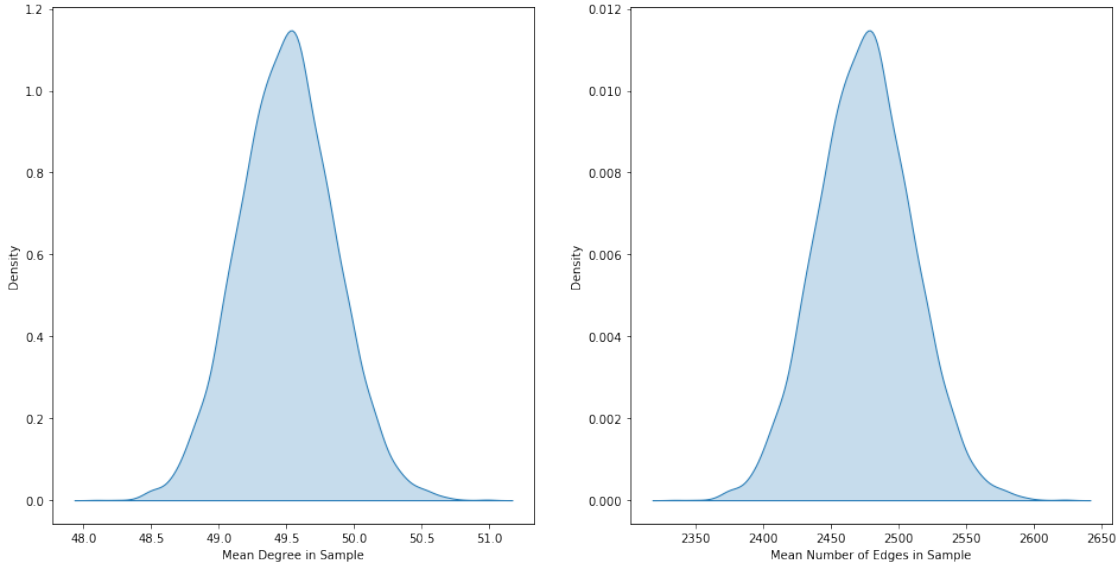


[40]:
```
display(np.mean(degrees), np.mean(num_edges))
```

```
499.44497999999993
```

```
249694.98
```

[41]:
```
degrees, num_edges = sample_ER_graphs(10000, 100)
```



[42]:
```
display(np.mean(degrees), np.mean(num_edges))
```

```
49.500533
```

```
2475.0533
```