Introduction

Permutation tests on networks when the nodal attributes are randomly assigned to the treatment group.

This tool is a strong opponent against the parametric tests proposed to solve the problem in this domain.

Randomization tests of no treatment effect are actually permutation tests for the independence between the treatment and the network.

The sharp null hypothesis is that the treatment assignment is completely independent of the network, in other words. The network would be exactly the same regardless of treatment assignment.

The test-statistic in these models somehow maps the permutated node label and the observed network to a scalar value.

Test statistics may rely on different parameters in the graph, for example, when we want to take the edge count as the test statistic, alternatives that change the counts of edges may affect the test statistic, but the changes in the higher order properties of the graph may have no contribute to the test statistic.

For example, we can define test statistics that are sensitive to the treatment effects that influence the clustering coefficient or even centrality in the graphs. The goal of this is indeed to detect the presence of treatment effects in the network.

In this model, observations are thought to have a set of fixed potential outcomes, each revealed by a particular treatment assignment. When treatment is assigned independently of the potential outcomes, statistical analysis reveals the causal effect of treatment.

Causal Inference for the networks

In this part, we assign nodes to different treatments denoted by Z_i, So if some node I has been selected for the treatment the z corresponding to that node is 1 and 0 if otherwise.
Also we denote the response variable corresponding to the outcome of that treatment with y_i or y_i(z_i). We detect the treatment effect whenever we observed at least one node with
y_i(z) != y_i(u), in this sense that the treatment effect was not same as some other treatment.

With the null hypothesis of no effect or in other words y(z) = y(u) for all the nodes. We also define test statistic over the distribution of different Zs keeping y fixed to be able to detect if an assignment Z, has a significant effect on the responses measured by our test statistic. Below, you can see how we define p-values in this context.

$$p^{+}(\mathbf{z}, \mathbf{y}) = \Pr(T(\mathbf{Z}, \mathbf{y}) \geq T(\mathbf{z}, \mathbf{y})) = \sum_{\mathbf{Z} \in \Omega} \Pr(\mathbf{Z}) I(T(\mathbf{Z}, \mathbf{y}) \geq T(\mathbf{z}, \mathbf{y})),$$

For example, a difference of means statistic would be sensitive to treatments that made the two groups different on average, but insensitive to treatments that only operated on the variance of the outcome.

Also keep in mind that, causal inference is only acceptable if we have random assignments in the permutation tests and not in other contexts.

In the context of social network, take we want to capture the treatment assignment in this way that W_ij = Zi + Zj
(either of the nodes being assigned would be enough to take the dyad selected)
And also the response variable here would be showing if there is an edge between the nodes i and j. With different test statistics here, we can detect the presence of different kinds of treatment effects in networks.

Local approaches

By local, we mean the test statistics that capture the effects between or within the treatment and control group for example, the number of edges between the treatment and the control group.

In the local approaches we defined the test statistic as combination of different statistic of different groups of treatment and control with linear and non-linear variations or even parameters from the distribution of the test statistic. **T_COD, T_PMF, T_CF.**

Global approaches

There are certain statistics that can capture the alterations in the global properties of the graph such as centrality scores and clusterings. In these kinds of statistics, we first, compute the test statistic without counting for the treatment and control group assignments, and then by takings them into account, we test the hypothesis of no effect with certain tests like fisher's exact test (this test assumes the same number of assignments in the treatment and control group when assigning nodes to different groups).

Simulations

In the simulations, we use 100 nodes with 50 of them assigned to treatment group and the rest assigned to the control group. We have used 500 replications and used the significance level of 0.05 to actually test the hypothesis of no effect. Besides, We computed the type 1 and 2 error by generating a graph and randomly assign groups, and by assigning groups and then generating graphs based on those treatment groups respectively.

Test Statistics:

* Edge Diff: computed by R2 / m2 - R0 / m0 to actually account for the edges within two groups in terms of their density.

* CF statistic by Chen and Friedman used in randomization tests

* CoD statistics in QAP regression for each of the classes W in [0, 1, 2]

* Fisher's exact test applied to the clusterings

* A Wilcoxon–Mann–Whitney test applied to nodal eigenvector centrality

For the sake of comparison with a parametric model, ERGM was used with a term accounting for GWESP (geometrically weighted edgewise shared partners) with a F-test to test the hypothesis of independence between Y and W.

Data generating process for the networks

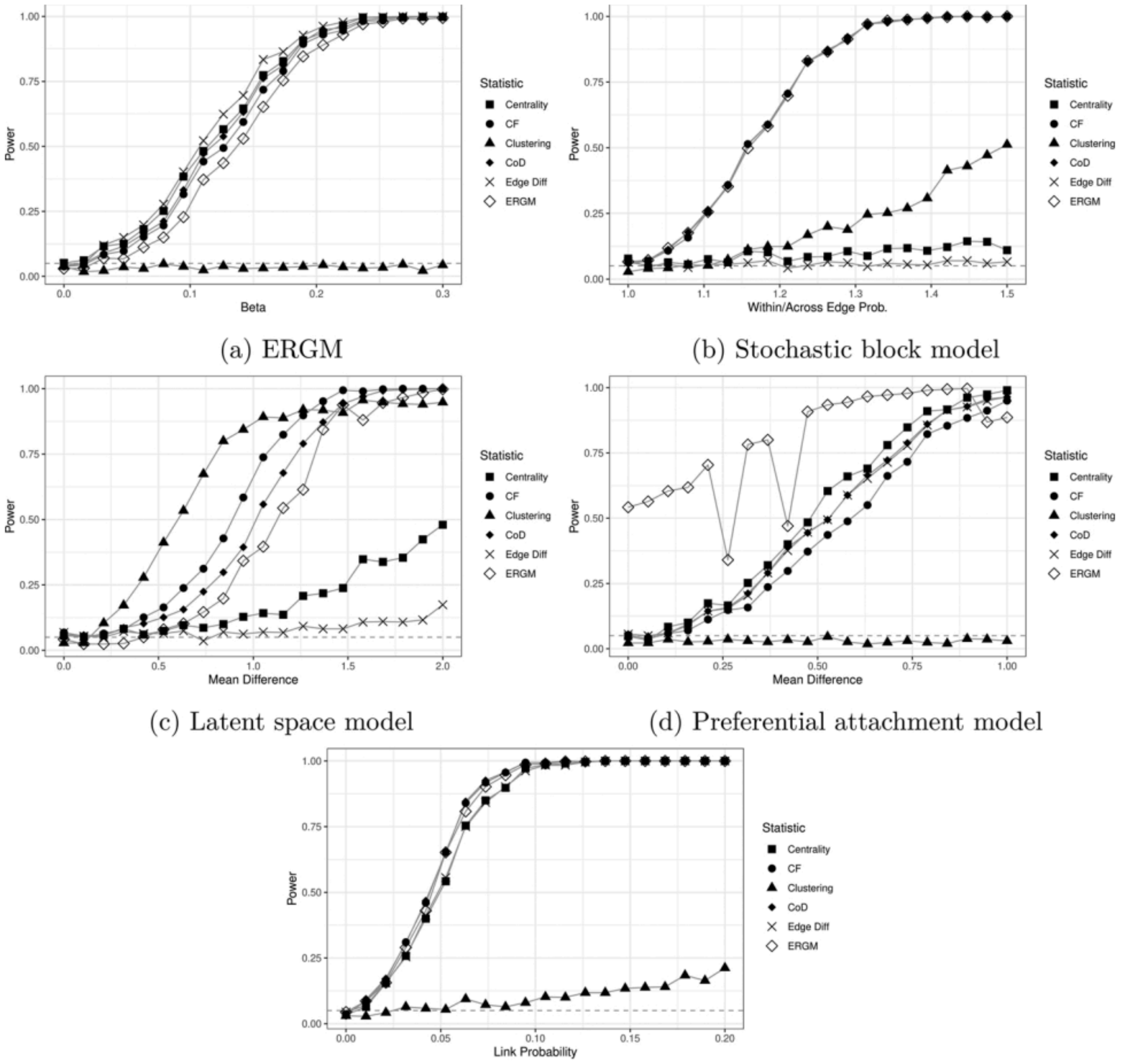
- An exponential random graph model in which treated units are more likely to form edges.

- A stochastic block model with blocks defined by treatment assignment with varying within and across block edge probabilities.

- A latent space model in which treatment status determines latent positions.

- A preferential attachment model in which treated units are more desired than control nodes.

- A process in which treated neighbors in an Erdös-Renyi graph are more likely to complete a triangle.



(a) ERGM

(b) Stochastic block model

(c) Latent space model

(d) Preferential attachment model

(e) Neighborhood closing model

The strength of all the models with different test statistics were shown in the above plot in terms of the proportion of rejection of null hypothesis of no effect.

Preferential attachment models

These models propose a way to generate edges one by one in a way that by adding each node, the edge between that node and some other node in the rest of the graph would be determined by what is the degree of that other node in the rest of the graph. In this way, the final graph would have a form of power law distribution in its degree distribution that is so common in the social networks applications. This way of distribution is so that lots of nodes have low levels of degree, while some distinct nodes have higher levels. We can interpret this kind of pattern as having influential nodes in our graph.

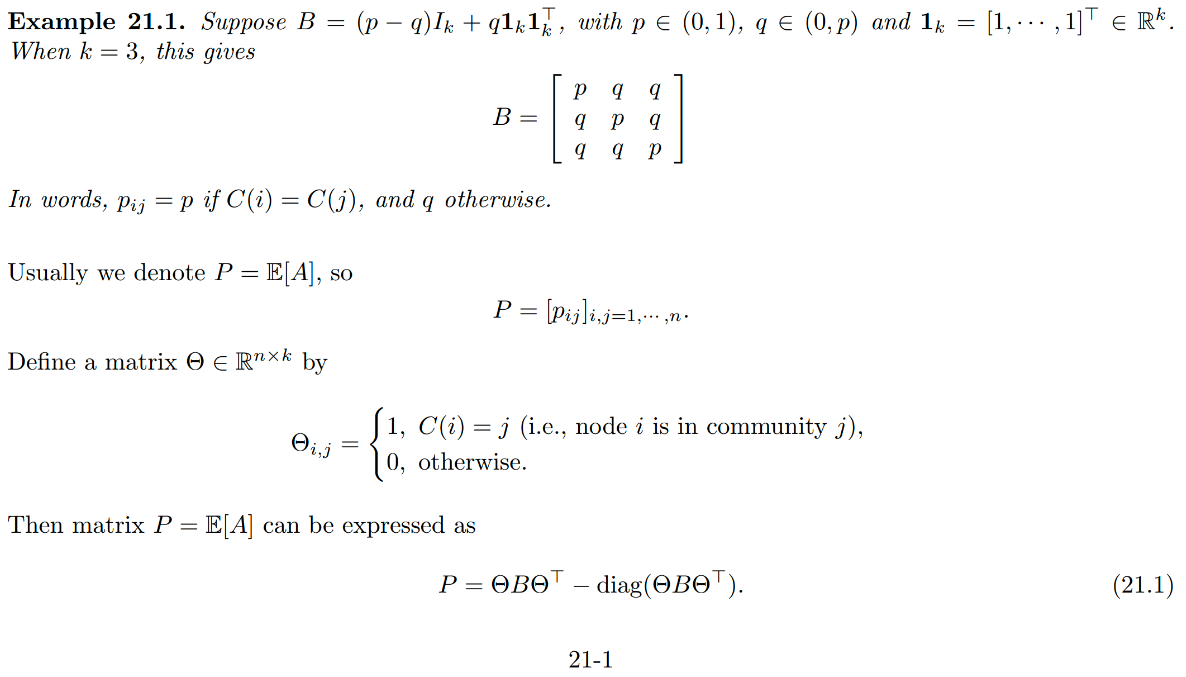
Latent Space Models

In the context of social networks, the latent space refers to some social space of unobserved latent characteristics that represent the potential transitive tendencies in the society. The probability of having a tie between two individual nodes is entirely dependent on those two nodes' positions in the latent space.

Euclidian Distance, logodds, MCMC

One of the approaches would be to encode the characteristics in a k-dimension vectors V_i. Those having similar characteristics would have more close angle between their vector of characteristics (Like what we have in the **embeddings**). Also, we can add a parameter for the rate of nodes activity rate to break the unidirectional relation between having directed ties in these kinds of networks.

Stochastic Block Models



Example 21.1. *Suppose* $B = (p-q)I_k + q\mathbf{1}_k\mathbf{1}_k^\top$*, with* $p \in (0,1)$*,* $q \in (0,p)$ *and* $\mathbf{1}_k = [1,\cdots,1]^\top \in \mathbb{R}^k$*. When* $k=3$*, this gives*

$$B = \begin{bmatrix} p & q & q \\ q & p & q \\ q & q & p \end{bmatrix}$$

In words, $p_{ij} = p$ *if* $C(i) = C(j)$*, and* q *otherwise.*

Usually we denote $P = \mathbb{E}[A]$, so

$$P = [p_{ij}]_{i,j=1,\cdots,n}.$$

Define a matrix $\Theta \in \mathbb{R}^{n\times k}$ by

$$\Theta_{i,j} = \begin{cases} 1, & C(i) = j \text{ (i.e., node } i \text{ is in community } j\text{),} \\ 0, & \text{otherwise.} \end{cases}$$

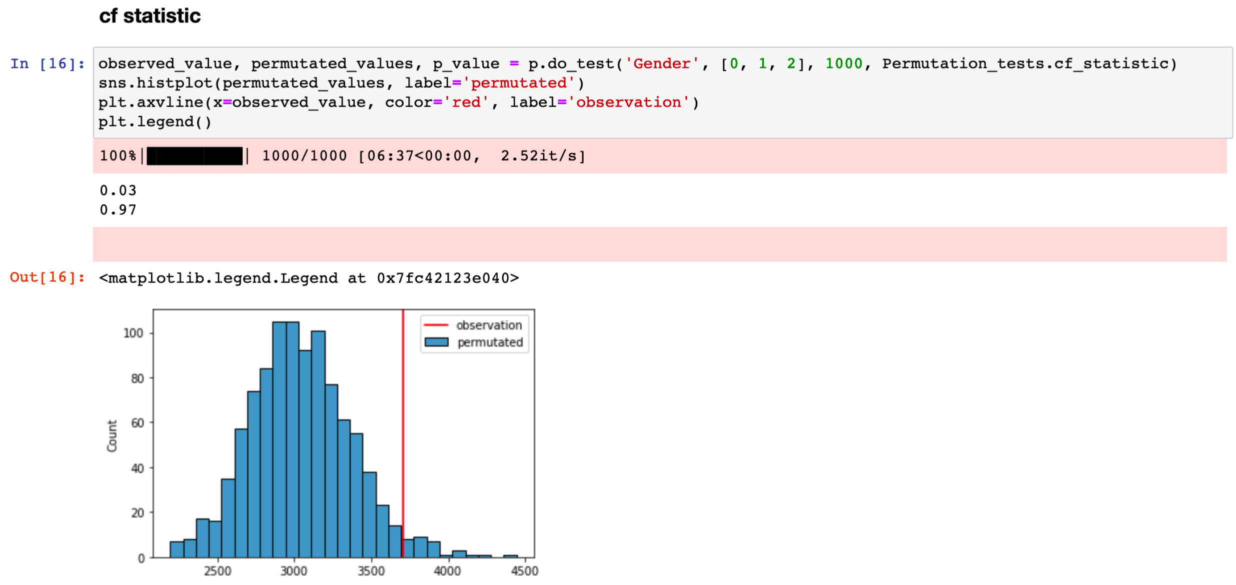
Then matrix $P = \mathbb{E}[A]$ can be expressed as

$$P = \Theta B\Theta^\top - \text{diag}(\Theta B\Theta^\top). \tag{21.1}$$

Models that have different rates of edge probabilities for the within-group and between-group ties. In other words, reduce the nodes and the probabilities between them to a lower dimensional matrix. As a result of that, the probability of having an edge between two edges would rely on the probability of the transformation of those two nodes using the map from n to k in other words we can reduce the adjacency matrix of size n * n to k * k with this transformation $p_{i,j} = B_{C(i),C(j)}$.

Examples:

Checking whether gender has anything to do with the Harry-Potter movies' characters interactions the first three test statistics introduced:



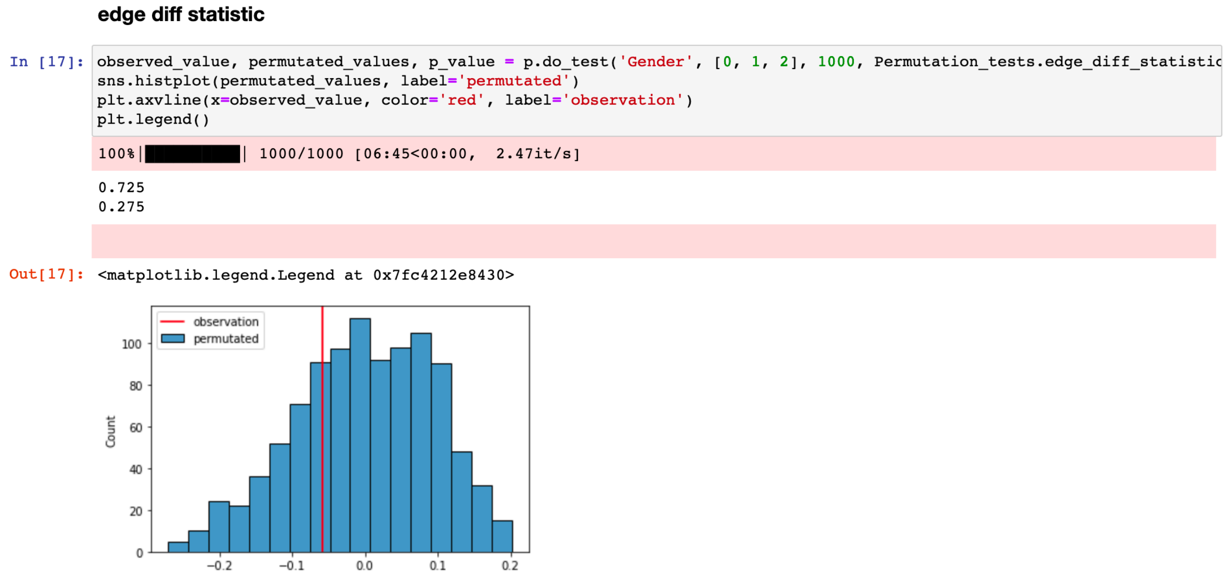
cf statistic

```
In [16]: observed_value, permutated_values, p_value = p.do_test('Gender', [0, 1, 2], 1000, Permutation_tests.cf_statistic)
         sns.histplot(permutated_values, label='permutated')
         plt.axvline(x=observed_value, color='red', label='observation')
         plt.legend()

100%|██████████| 1000/1000 [06:37<00:00,  2.52it/s]

0.03
0.97

Out[16]: <matplotlib.legend.Legend at 0x7fc42123e040>
```



edge diff statistic

```
In [17]: observed_value, permutated_values, p_value = p.do_test('Gender', [0, 1, 2], 1000, Permutation_tests.edge_diff_statistic
         sns.histplot(permutated_values, label='permutated')
         plt.axvline(x=observed_value, color='red', label='observation')
         plt.legend()

100%|██████████| 1000/1000 [06:45<00:00,  2.47it/s]

0.725
0.275

Out[17]: <matplotlib.legend.Legend at 0x7fc4212e8430>
```

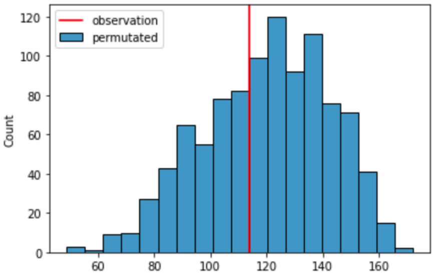
normal statistic

```
In [18]: observed_value, permutated_values, p_value = p.do_test('Gender', 0, 1000, Permutation_tests.within_group_edges_statisti
         sns.histplot(permutated_values, label='permutated')
         plt.axvline(x=observed_value, color='red', label='observation')
         plt.legend()
```

```
100%|██████████| 1000/1000 [02:10<00:00,  7.65it/s]

0.621
0.379
```

```
Out[18]: <matplotlib.legend.Legend at 0x7fc4211eb370>
```



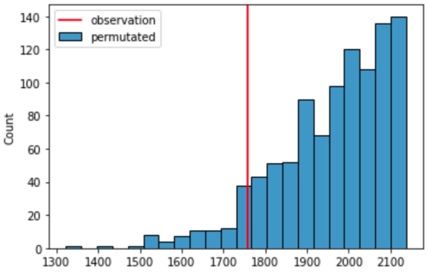
Mannwhitneyu Test

```
In [73]: observed_value, permutated_values, p_value = p.do_test('Gender', None, 1000, Permutation_tests.mannwhitneyu_test)
         sns.histplot(permutated_values, label='permutated')
         plt.axvline(x=observed_value, color='red', label='observation')
         plt.legend()
```

```
100%|██████████| 1000/1000 [00:27<00:00, 36.57it/s]

0.916
0.084
```

```
Out[73]: <matplotlib.legend.Legend at 0x7fc423606370>
```


observation
permutated
Count