Assignment与心型性的性 医神经 编号 Statistical Analysis

Spring 2024

Overview

The objective of this description and network statistical analysis. This assignment covers topics in lessons 12 and 13.

Submission WeChat: cstutorcs

Please submit your Jupyter Notebook A5-YOURGTUSERNAME.ipynb with requirements.txt so that we may be able to replicate your Python dependencies to run your code as needed. ASSIGNMENT Project Exam Help

With Anaconda, you can do this by running:

conda list -e > requirements.txt

Ensure all graphs and plots are properly labeled with unit labels and three for x & y axes.

Producing readable, interpretable graphics is part of the grade as it indicates understanding of the content – there may be point deductions if plots are not properly labeled.

Getting Started

In this assignment you will use football on the NCAA College Football Network. For the Hierarchical Random Graph in Part 2, you will be using the modified file "football-hrg.gml" and PyHRG. For part 3 you will be using the "slashdot.txt" file. The graph loading code is already provided for you. You **DO NOT** need to alter it.

NCAA College Football Network

Part 1: Structural Properties of the Graph [25 points]

The goal for the first part of this assignment is to show the structural properties of the empirical network. Just like the previous assignment, the code to load the graph is already provided and you don't need to make any modifications.

 Complete the calculate_graph_statistics function to calculate the network diameter, characteristic path length (CPL), average clustering coefficient, transitivity, assortativity, and degree sequence (a list of all the degrees for each node). Return a dictionary of the results with the name of each statistic as the ey. CS in the sequence (a list of all the degrees for each node). Return a dictionary of the results with the name of each statistic as the eye. CS in the sequence of the results with the name of the sequence of the results with the name of the sequence of the results with the name of the sequence of the results with the name of the sequence of the sequen

2. Complete the within the grave within

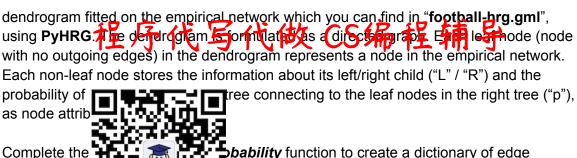
Additionally, complete the **plot_nmi_vs_resolution** function to display the NMI for each resolution as time plot hat: CSTUTOTCS

- 3. Complete the *calculate_best_partition* function to find the resolution associated with the partition that generates the highest NMI.
 - Additionally, complete he the plactition function for the best partition assignments. Plot both the ground truth community assignments and the best partition assignments. Plot both networks side by side as subplots within a single figure. Make sure to label which graph is which flow air we ground truth res at 163.com
- 4. Complete the *calculate_inter_community_density* function to calculate the intercommunity connection density matrix (see matrix P in L12: Generating Networks with Community Structure of the definite value of the definite value.
 - Additionally, complete the *plot_p_matrix* function to plot the inter-community detection density matrix as a heatmap. Make sure to annotate the values within each cell and provide a legent tps://tutorcs.com
- 5. Run the code in the 1.5 cell. How does the resolution impact the NMI? Is the partition for the best NMI a good match to the ground truth? Justify your answer based on the visual plot and the NMI value itself.

Part 2: Graph Generation [40 points]

In this section you will use different graph generators that you learned about in the lessons to generate graphs based on the statistics you calculated in part 1.

- Complete the generate_configuration_graphs and generate_sbm_graphs functions
 to generate n_graphs=100 graphs using the Configuration Model and Stochastic Block
 model graph generator in NetworkX (with the configuration_model function, use the
 create_using=nx.Graph() argument to get a Graph and not MultiGraph).
- 2. Here you will be working with the Hierarchical Random Graphs. We have composed a



Complete the **Complete** the **Complet**

Then, comple **probe** function to generate a networkx graph based on the edge probabilities calculated in the previous part. Specifically, if two nodes *i* and *j* have probability *p* of having an edge, generate an edge randomly with probability *p* for each pair of nodes. **CStutorcs**

Finally, complete the <code>generate_hrg_graphs</code> function to generate n_graphs=100 HRG graphs using these edge probabilities and the generate function you just completed above. This should function you just completed your own <code>generate_graph_from_probs</code> inside the loop instead of calling a network generator.

Email: tutorcs@163.com

3. Complete the *calculate_generated_statistic* function to calculate the network diameter, CPL, average clustering coefficient, transitivity, and assortativity for each graph in a list of graphs. For nat the return as a dictionary where the key is a string describing the property and the value is a list of the result for each of the graphs in the list. Note if any of your graphs are not connected, you may use the largest connected component.

Next, complete the campare generated to ground truth function to compare the diameter, CPL, average clustering coefficient, transitivity, and assortativity for the empirical network with the sampled values from each of the graph models using a one-sample t-test.

Additionally, complete the *plot_graph_statistics* function to visualize the distribution of each individual network property (e.g. CPL, average clustering) as a boxplot. Create the plot as a single plot with a series of five side-by-side boxplots (one for each property) as subplots. Label the overall plot with the generator name and label each individual box plot with the property being shown along the y axis.

4. How do the statistics compare against the three different types of graph generators? Are there any particular measures that are very different? Can you explain how the algorithm for each generator influences the statistics that are different? Which graph generator would you say is the best fit for the ground truth? Justify your answer based on the t-test results and the box plots.

Slashdot Ne裡呼代写代做 CS编程辅导

Part 3: Sampling [35 points]

In this part, we will be show should show should be shown the list of edges in this show the list of edges in the list of edges

Additionally, complete the **plot_estimate_histogram** function to plot a histogram of the estimated number of nodes in the network (the y-axis is the frequency of occurrence and the x-axis is the estimated in the plot in th

rinally, complete the **plot_sample_size_error** function to alor a line plot of the estimated number of nodes abbidith erlocked presenting the standard deviation for each sample size. The line plot value will be the mean over the 1000 trials for a given sample size and the error bars the standard deviation for that sample size. Along with this plot the true_size is notizental free deviation for that sample sizes.

- 2. Complete the **estimate_edges function** to estimate the number of edges using the induced subgraph sangling and the order in the network. Repeat this n_iter=100 times and save the resulting edge count estimates in a list.
 - Additionally, complete the **plot_edge_estimate_distribution** function to plot a histogram of the estimates of the number of edges along with the ground truth (the y-axis is the frequency of occurrence and the x-axis is the estimated number of edges).
- 3. Run the code in cell 3.3. How does the estimate of the number of nodes change as you increase the sample size? What is the smallest sample size you would consider a good choice for estimating the number of nodes?