### Analysis 6: Assortativity Instructions

### Construct your solution using Analysis-6-Assortativity-Lastname.Rmd

### Final version due in Laulima evening of Monday April 4th.

### 60 points available (indicated allocation to subquestions may be modified)

It is simple to compute assortativity, so this could have been a very short assignment consisting of just a few assortativity expressions. However, the real work of data science is often in preparing the data and in interpreting the results in terms of domain processes. This assignment includes data preparation, and we apply analytics (models and metrics) we previously learned to understand these new networks, making this a kind of “midterm exam” review! Working with assortativity also prepares us for next week's topic on community detection.

### Prelude: Primary School Contact Network

We will be working with a contact network of primary school students and their teachers gathered by RFID tags. To begin, go to:

external link: <http://www.sociopatterns.org/datasets/primary-school-cumulative-networks/>

and read about the data. (I also recommend that you watch the animation embedded in their paper. Can you tell when they have recess, and then return to their classes? Which classes seem to have more of an interactive learning style? We will get into temporal analysis briefly later this semester.)

Then **download both data files** (but not the metadata, as it is already included in the graphs)**:**

* **Cumulative network day 1, GEXF format, 44 KB**
* **Cumulative network day 2, GEXF format, 57 KB**

In addition to having you practice assortativity analysis, part of this assignment is intended to help develop your skills in obtaining and converting data. In particular, there are some issues with this data and with our analysis:

* First we have to unzip the files and fix the file extension.
* They are in gexf format, not readable in igraph, so we convert to graphml via Gephi.
* We want to combine these two days of data into one network, summing the contact counts and durations on the edges. We can merge graphs in Gephi, but Gephi “Sum” strategy only sums weights, and overwrites count and duration data!
* We are interested in assortativity of students, but there are teachers in the network we will have to remove.
* The assortativity metrics we will use do not take weights into account. Also, we want to filter out edges with counts and durations that indicate very brief contacts that may not represent true interaction (a tag records 20 seconds of contact even if students pass by each other for 1 second).
* We would like to analyze by grade level, but grade is coded only indirectly as classname (e.g., "2A" and "2B" are the same grade level but different classes: we want to extract "2" as the value for grade).

Also, before we start analyzing a network we should become familiar with its basic properties. You'll start with the basic metrics and random models we learned in previous weeks.

Instructions are in the .Rmd. I add a few comments below.

### 1. Preparing the Network (15 pts)

**(a) Conversion and Simplification.** After uncompressing the files and fixing the extension, load one into a Gephi workspace and then load the second into the same workspace with Don’t Merge as the strategy. (I tested Edge merge strategy “Sum” and it only sums the weights, not the count and duration attributes.) Then write out as .graphml without the include Gephi layout attributes. Load the result into igraph and use summary to show that it has been loaded with the relevant attributes.

Then simplify, with edge.attr.comb being set to sum duration and count as well as weight. We save this into a new variable PS\_v2 so you can compare the versions and repeat things if needed. Check that you have the expected number of vertices and edges and that the max values are as expected.

**(b) Filtering Teachers.** This is straightforward with delete\_vertices on the “Teachers” classname. Check vertex and edge counts again. We save into PS\_v3.

**(c) Filtering Incidental Contacts.** Now we make the final version called PSF by filtering contacts that were too brief to be considered substantial interaction. This removes a large number of edges that would otherwise be noise (results are muddied when they are left in), especially given that the assortativity metrics don’t consider weights, so these low duration contacts would have equal impact as long duration ones. The .Rmd template discusses the rationale for which values are filtered. Other policies are defensible and I’ve tried various policies without significant change to the results as long as the large counts of low duration interactions are removed. However, it will facilitate our grading if everyone uses the same filtering policy. We do the deletions in two steps to keep the logic simple, using tests on duration and count.

**(d) Adding Grade Attribute.** Make a new attribute called 'grade' and give it appropriate values by conversion of the 'classname' attribute. *Hint*: substr extracts substrings.

### 2. Investigating the Network (20 pts)

Henceforth we work only with PSF, the filtered graph that does not have teachers or low frequency or duration contacts.

**(a) Random Models.** Make a random G(n,m) graph model and a randomly rewired configuration model for PSF. We use these to check what effects are random or due to the degree sequence. (We do not use the configuration model, as we need the vertex attributes to compute nominal assortativity.)

**(b) Transitivity.** Make a table of global transitivity for PSF and the two random models. Also compute edge density to aid interpretation. (Density is the probability that there is an edge between any given pair of vertices, and is |E| divided by maximum possible number of edges.)

Discuss: Is the global transitivity typical of social networks? How much of the transitivity is accounted for by density of random connections? By the degree distribution?

**(c) Degree Distribution.** Plot the degree distribution of PSF using both lin-lin and log-log axes. Nothing new here: you know how to do it.

Discuss: How would you characterize the distribution? Does it have a scale free or heavy tailed structure typical of social networks, or something else? Based on the plot, do you think preferential attachment is operating with respect to the **number of persons contacted** (degree), or is this random or affected more by other factors?

**(d) Attribute Distributions.** Plot the **count** and **duration** of contact by converting their frequency tables into vectors and making two plots. This takes more work to make a nice plot with circles for data points. Make sure that your x axis is based on the values being tabulated, not the index into the vector! The x axis values should go up to the max values computed in 1c.

Discuss: How would you characterize these distributions? Do they have a scale free or heavy tailed structure typical of social networks, or something else? What can you conclude about the behavior of students using these plots? (Careful: they are not degree distributions.)

### 3. Assortativities (25 pts)

Interactions between primary school students may be shaped partly by the environment: students presumably are more likely to be nearby or interact with those in the same classroom. The data may also reflect choices made by the students within and outside the classroom, e.g., to associate with students according to gender, grade level, or popularity (degree measures). Here we use the degree and nominal assortativity metrics to compare the predictive values of these attributes. We compare to the Configuration model (rewired graph) to see whether each given assortativity is structural. (We do not compare to G(n,m) as that model does not have attributes.)

**(a) Assortativity Computations.** Prepare a table of the following form:

| Natural PSF | Degree Assortativity | Nominal Gender Assortativity | Nominal Class Assortativity | Ordinal Grade Assortativity |
| --- | --- | --- | --- | --- |
| Rewired PSF | Degree Assortativity | Nominal Gender Assortativity | Nominal Class Assortativity | Ordinal Grade Assortativity |

For nominal assortativity, you'll need to first encode them as integers starting with 1.

**(b) Plotting Strongest Assortativity.** Just to demonstrate the assortativity visually, and give you a little more plotting experience, plot the grade values for the vertices at the head\_of and tail\_of ends of the edges against each other, e.g., head\_of on x axis and tail\_of on y axis. You may take any approach that makes a nice plot, for example: plot with jitter (adjust the factor) as done in class; smoothScatter (which provides a heat map); or ggplot geom\_bin\_2d if you are familiar with it. I think smoothScatter will be the easiest way to get a nice result.

**(c) Network Visualization.** Here we take a quick trip to Gephi to visualize the structure of the contact network and color it by the top two assortative attributes, classname and grade, to see how they correspond to the actual structure. These are of interest because grade was derived from classname.

We do not need high resolution images, and we don’t need labels (we’ll get into more detail next week). Just give it a clear layout and use the Gephi “camera” to make two .png images to include in the .Rmd document.

**(d) Interpretation.** Now we can discuss all the above results to come up with a coherent interpretation.

First address whether any of the assortativities for degree, gender, grade, and classname are structural.

Then interpret the results in terms that would make sense to a client such as educators running this school. What do the results say about the various factors affecting how students associate with each other? Reference any relevant results from the previous analyses, but try to translate these results into an explanation in terms of the domain, not just the graph.

### Finishing It

Be sure to start a fresh R session and run Knit to HTML to turn in a .html. Check the numbers for correctness after it has run. The .nb.html that is made by Preview may not be sufficient. Preview shares the same environment as your console, and re-executing earlier code chunks may make use of later versions of the network. I tried to reduce the danger of this by using version numbers PS\_v1 etc, as at one point when I was using only one graph the earlier tables were printing the later numbers, but you should always generate a final report with Knit.