Technical Review: Student-Course Network Analysis Proposal

Kelly Gothard

UVM Office of Institutional Research

**Dataset**

The data required for this project are each individual students’ courses per semester, and their college and major. Each student will have taken n courses for m semesters, so there will be n\*m rows per student. A student can take the same course in different semesters, but cannot take the same course twice in one semester.

We cannot aggregate students to de-individualize the dataset because each student has a unique course sequence, and we need to be able to map students to the courses they took at a particular time and associate students via mutual courses taken. However, identifiers such as name, UVM ID, etc, will be excluded from the dataset. The observations made during this study will be regarding communities of un-identifiable students and distributions of student connectivity.

**Software**

To construct a network data structure, detect communities, create visualizations, and compute measures of connectivity, programs will be written in Python 3.7.1. We anticipate the use of open-source Python libraries found in Table 2, and such dependencies will be included in a transportable Python environment upon conclusion of this project so that it may be used in the future.

To activate on another machine:

cd Student-Course-Network

source student\_network\_venv/bin/activate

https://docs.python-guide.org/dev/virtualenvs/

**Computing Resources**

Large networks, such as the student-course connectivity network that we will be working with in this study, can require supplemental computing resources to maintain a reasonable timeline of computational processes. For high-volume computational tasks such as community detection, we will utilize the Vermont Advanced Computing Core.

**Figures**

Table 1. Dataset Schema Example

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Individual Identifier (ID) | College1 | College2 | Major1 | Major2 | Course | Semester |
| Student1 | CEMS | CEMS | DS | DS | CS 021 | F 2018 |
| Student1 | CEMS | CEMS | DS | DS | MATH 121 | S 2019 |
| Student2 | CALS | CAS | CDAE | PSYS | PSYS 001 | F 2018 |

Table 2. Open-Source Python Libraries

|  |  |
| --- | --- |
| Library Name and Website | Description |
| Pandas  <https://pandas.pydata.org> | pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. |
| NetworkX  <https://networkx.github.io> | NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. |
| NumPy  <https://numpy.org> | NumPy is the fundamental package for scientific computing with Python. It contains among other things:  a powerful N-dimensional array object  sophisticated (broadcasting) functions  tools for integrating C/C++ and Fortran code  useful linear algebra, Fourier transform, and random number capabilities |
| MatPlotLib  <https://matplotlib.org> | Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits. |
| Community  <https://python-louvain.readthedocs.io/en/latest/> | This module implements community detection.  It uses the louvain method described in Fast unfolding of communities in large networks, Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Renaud Lefebvre, Journal of Statistical Mechanics: Theory and Experiment 2008(10), P10008 (12pp). |