Ryad GUENNOUN : https://github.com/Kirabsol/RGNET4103

# NET4103 Project

## Description of the network dataset

The Facebook100 dataset [2] contains an anonymized snapshot of the friendship connections among n = 1, 208, 316 users affiliated with the first 100 colleges admitted to Facebook, all located in the United States. This comprises a total of m = 93, 969, 074 friendship edges (unweighted and undirected) between users within each separate college.

Each vertex is associated with an array of social variables representing the persons status (undergraduate, graduate student, summer student, faculty, staff, or alumni), dorm (if any), major (if any), gender (M or F), and graduation year. Across all networks, only 0.03% of status values are missing. Other variables have slightly higher missing rates (gender: 5.6%; graduation year: 9.8%). Dorm and major have higher rates still, which is likely related to off-campus living and undeclared majors. The completeness of these data reflects the pervading social norms surrounding data privacy expectations in 2005, and possibly a selective bias toward users who disliked the default setting of sharing all information within the college network.

For nearly all colleges, alumni made up about 10-25% of users, a number that increased with the age of the network. Vertices labeled as faculty, staff or students who were not regular undergraduates (graduate students and summer students) made up on average 4.1% of each population.

Each college network includes an index variable that gives its ordinal position of when it joined Facebook: Harvard is 1 and Trinity College is 100 (Fig. 1a). For each network, we acquired college-level variables (enrollment, public vs. private, semester vs. quarter calendar) from the Integrated Postsecondary Education Data System (IPEDS) provided by the U.S. Department of Education. Full-time undergraduate enrollment from 2007, the earliest date for which data are fully available, was used as a proxy for 2005 enrollment.

By dividing the number of undergraduate accounts in each college network by reported enrollment, we can estimate the fraction of students in each network who were on Facebook, a measure of service adoption. In some cases, the estimated ratio exceeds 1.0 as a result of either errors in our enrollment numbers, part-time students on Facebook who were not counted as full-time enrolled, or multiple/fake accounts at the few colleges that allowed students to control multiple email aliases and circumvent Facebooks initial limits on access.

## Question 1: Reading

Read the following documents [3, 2, 1]

## Question 2: Social Network Analysis with the Facebook100 Dataset

The smallest network (Caltech) has 762 nodes in the largest connected component (LCC), and the largest has more than 40000 nodes in the LCC.
Let's use three networks from the FB100: Caltech (with 762 nodes in the LCC), MIT (which has 6402 nodes in the LCC), and Johns Hopkins (which has 5157 nodes in the LCC).

**(a)** *(1 point)* For these three networks plot the degree distribution for each of the three networks that you downloaded. What are you able to conclude from these degree distributions?

Caltech has a much smaller LCC, yet still shows significant variance in degree. This suggests presence of hub nodes, though the tail drops off a bit more quickly than in the other networks.

MIT's network shows a more robust power-law behavior. The slow decay implies stronger heterogeneity in connectivity.

The Johns Hopkins network resembles MIT's, but with fewer extreme hubs. This could imply a slightly less pronounced scale-free structure or differences in community structure or user behavior.

**(b)** *(1 point)* Compute the global clustering coefficient and mean local clustering coefficient for each of the 3 networks. In addition compute the edge density of each network. Should either of these networks be construed as sparse?
Based on the density information and the clustering information what can you said about the graph topology?

All 3 networks are sparse (Edge Density << 1), but Caltech's is less sparse than the others. There is local cohesiveness: friends of a user are often also friends with each other. These networks have high clustering and short paths.

**(c)** *(1 point)* For each network, also draw a scatter plot of the degree versus local clustering coefficient.
Based on these calculations as well as your previous ones, are you able to draw any conclusions about any similarities or differences between the tree networks? What other observations can you make?

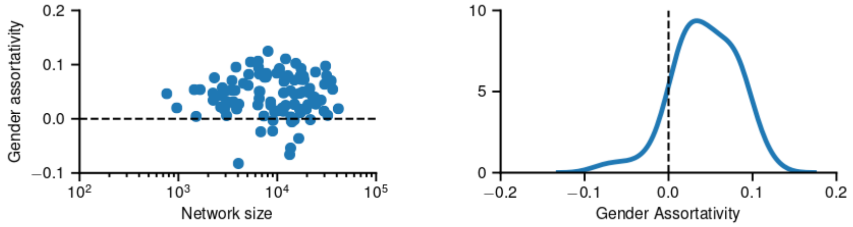
We notice, for all 3 network graphs, that as the nodes' degree raises, the clustering coefficient diminishes strongly, whereas for low-degree nodes, the clustering coefficient can reach all values from 0 to 1. It's clear that MIT's and John Hopkins' network are very similar, and differ in values from Caltech's.

## Question 3: Assortativity Analysis with the Facebook100 Dataset

In this question we expect you will compute the assortativity on a large set of graphs (if possible all the graphs).

**(a)** *(2 points)* Of the FB100 networks, investigate the assortativity patterns for five vertex attributes: (i) student/faculty status, (ii) major, (iii) vertex degree, and (iiii) dorm, (iiiii) gender. Treat these networks as simple graphs in your analysis.
For each vertex attribute, make a scatter plot showing the assortativity versus network size n, with log-linear axes for all 100 networks, and a histogram or density plot showing the distribution of assortativity values. In both figures, include a line indicating no assortativity. Briefly discuss the degree to which vertices do or do not exhibit assortative mixing on each attribute, and speculate about what kind of processes or tendencies in the formation of Facebook friendships might produce this kind of pattern.

For example, below are figures for assortativity by gender on these networks. The distribution of points spans the line of no assortativity, with some values nearly as far below 0 as others are above 0. However, the gender attributes do appear to be slightly assortative in these social networks: although all values are within 6% in either direction of 0, the mean assortativity is 0.02, which is slightly above 0. This suggests a slight amount of homophily by gender (like links with like) in the way people friend each other on Facebook, although the tendency is very weak. In some schools, we see a slight tendency for heterophily (like links with dislike), as one might expect if the networks reflected heteronormative dating relationships.



## Question 4: Link prediction

In this question we expect you will compute the link prediction algorithms on a large set of graphs (> 10).
**(a)** Read the following documents [4].

**(b)** *(2 points)* Implement the following link prediction metrics: common neighbors, jaccard, Adamic/Adar. We use the scikit-learn API as an example for our implementation of the link prediction metrics. Please use the implementation (in listing. 1) as an example. Your implementation should inherit from the class LinkPrediction defined in listing. 1. You should implement yourself the given metrics, don't used the ones defined in Networkx.

**(c)** *(2 points)* Evaluating a link predictor:

1. Select graph $G_{fb}$(V, E) in the Facebook100 dataset

2. Randomly remove a given fraction f ∈ [0.05, 0.1, 0.15, 0.2] of edges $E_{removed}$ from the original graph $G_{fb}$.

3. For each node pair in the graph |V |×|V |, for each node pair compute the link

predictor metrics of interest p, these are the predicted "friendship" $E_{predict}$.

4. Sort in decreasing order of confidence as a function p from the node pair $E_{predict}$ and then we take the first k pairs of nodes $E_{predict}^{(top@k)}$.

5. Compute the size of the intersection between the edge set of removed edges and the edge set of predicted node $|E_{removed} \cap E_{predict}^{(top@k)}|$ . Then compute the top@k, recall@k and precision@k (for k = 50, 100, 200, · · · , 400) using the k best scored edges provided by link predictor algorithm (see [5] for more information). Where the top@k predictive rate is the percentage of correctly classified positive samples among the top k instances in the ranking produced by a link predictor P.

$$Precision = \frac{|TP|}{|TP| + |FP|}$$

$$Recall = \frac{|TP|}{|TP| + |FN|}$$

Prediction Terminology: TP stands for true positives, TN stands for true negatives, FP stands for false positives, and FN stands for false negatives.

**(d)** *(2 points)* Choose a couple of graphs in the facebook100 dataset run and evaluate each link predictor on them, and conclude on the efficiency of the following metrics: common neighbors, jaccard, Adamic/Adar.

```python
from abc import ABC
from abc import abstractmethod
import networkx as nx
import numpy as np
import progressbar

class LinkPrediction(ABC):
    def __init__(self, graph):
        """
        Constructor

        Parameters
        ----------
            graph: Networkx graph
        """
        self.graph = graph
        self.N = len(graph)

    def neighbors(self, v):
        """
        Return the neighbors list of a node

        Parameters
        ----------
            v: int
                node id

        Return
        ------
            neighbors_list: python list
        """
        neighbors_list = self.graph.neighbors(v)
        return list(neighbors_list)

    @abstractmethod
    def fit(self):
         raise NotImplementedError("Fit must be implemented")

class CommonNeighbors(LinkPrediction):
    def __init__(self, graph):
        super(CommonNeighbors, self).__init__(graph)
```

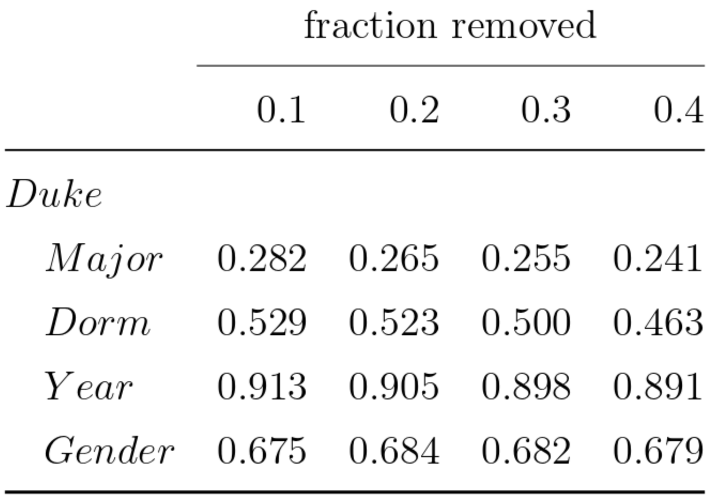
Listing 1: Python implementation example of a Link prediction metric

## Question 5: Find missing labels with the label propagation algorithms

In this question we expect you will compute the label propagation algorithm on a large set of graphs (> 10). We studied in class two algorithms with the name ”label propagation” that have different objective, choose wisely the one to implement.
**(a)** Read the following document [6].

**(b)** *(2 points)* Implement in python the label propagation algorithm [6], please consider pytorch and network for the development of your algorithm.

**(c)** *(2 points)* Choose a network from The Facebook100 dataset and randomly select 10%, 20%, and 30% of of the node attributes of the network to be removed.
Use the label propagation algorithm you implemented to recover the missing attributes. Perform this operation for each of the following attributes : ”dorm”, ”major”, ”gender”.



| | fraction removed | | | |
|---|---|---|---|---|
| | 0.1 | 0.2 | 0.3 | 0.4 |
| *Duke* | | | | |
| *Major* | 0.282 | 0.265 | 0.255 | 0.241 |
| *Dorm* | 0.529 | 0.523 | 0.500 | 0.463 |
| *Year* | 0.913 | 0.905 | 0.898 | 0.891 |
| *Gender* | 0.675 | 0.684 | 0.682 | 0.679 |

Table 1: Accuracy of the label propagation algorithm

**(d)** *(1 point)* For each case of the following percentage of missing attributes: 10%, 20% and 30% and for each of the following attributes: the ”dorm”, ”major”, ”gender” show the mean absolute error and accuracy score (as defined in eq. 1) of the label propagation algorithm as in the example provided in Table 1 for the Duke University Facebook network. Note we can use the formula eq. 1 for computing the accuracy. However, a better approach would have been to compute the F1-score [7].

$$accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \mathbf{1}(\hat{y}_i = y_i) \qquad (1)$$

## Question 6: Communities detection with the FB100 datasets

Formulate a research question about group formation among students in the FB100 dataset. To validate your hypothesis, use only a few universities and a community detection algorithm of your choice to extract the different groups of students. To help you formulate a research question, some of the following references might be useful [8, 9] and section 3.4 in [2].

**(a)** *(1 point)* Formulate a research question about group formation in FB100 and explain your hypothesis.

**(b)** *(1 point)* Write the code to validate your research question and show the result using a few selected community detection algorithms and graphs.

**(c)** *(1 point)* Explain the results and conclude whether your experiment confirms your initial hypothesis.

## References

[1] Jacobs, A. Z., Way, S. F., Ugander, J. & Clauset, A. Assembling thefacebook: Using heterogeneity to understand online social network assembly. In Proceedings of the ACM Web Science Conference, WebSci ’15, 18:1–18:10 (2015). URL https://arxiv.org/abs/1503.06772.

[2] Traud, A. L., Kelsic, E. D., Mucha, P. J. & Porter, M. A. Comparing community structure to characteristics in online collegiate social networks. SIAM Review 53, 526–543 (2011). URL https://arxiv.org/abs/0809.0690.

[3] Traud, A. L., Mucha, P. J. & Porter, M. A. Social structure of facebook networks. Physica A: Statistical Mechanics and its Applications 391, 4165 – 4180 (2012). URL https://arxiv.org/abs/1102.2166.

[4] Liben-Nowell, D. & Kleinberg, J. The link prediction problem for social networks. In Proceedings of the Twelfth International Conference on Information and Knowledge Management, CIKM ’03, 556–559 (2003). URL https://www.cs.cornell.edu/home/kleinber/link-pred.pdf.

[5] Yang, Y., Lichtenwalter, R. N. & Chawla, N. V. Evaluating link prediction methods. Knowledge and Information Systems 45, 751782 (2014). URL http://dx.doi.org/ 10.1007/s10115-014-0789-0.

[6] Bhagat, S., Cormode, G. & Muthukrishnan, S. Node Classification in Social Networks. In Social Network Data Analytics, 115–148 (Springer US, 2011). URL https://arxiv.org/abs/1101.3291v1.

[7] Mishra, A. Metrics to evaluate your machine learning algorithm (2018). URL https://bit.ly/2JqbRrt.

[8] Lee, C. & Cunningham, P. Community detection: effective evaluation on large social networks. Journal of Complex Networks 2, 19–37 (2013). URL https://doi.org/10.1093/comnet/cnt012.

[9] Sung, Y.-S., Wang, D. & Kumara, S. Uncovering the effect of dominant attributes on community topology: A case of facebook networks. Information Systems Frontiers 20, 10411052 (2016). URL http://dx.doi.org/10.1007/s10796-016-9696-0.