analysis_EMB_ex34_reproduce-d-classifier

April 18, 2023

1 Reproducing prior "D"-classifier with current workflow

1.1 Experiment description

1.1.1 Narrative set-up

Recent results suggest that our current methodology is unable to reproduce the prior results utilizing only degree information (ex33v2.1, see Reference Figure (1)). However, this does not rely on embedding information at all and hence, under logistic regression, should be able to reproduce prior "D"-classifier results nearly exactly, up to very minor fluctuations due to random observations. Hence, we find ourselves needing to re-examine the fundamentals of our current workflow's reconstruction.

1.1.2 Goal

The main goal of this notebook is to examine where the discrepancy in our current results and our previous "D"-classifier results are occurring. This notebook will be limited to exploring configuration models - any actionable insights will be discussed before similar notebooks are created to examine the LFR setting, or (hopefully), we adjust the current code base for more trustworthy results.

2 Set-up

2.1 Package management

2.2 Global config

---

3 Experiment

3.1 Data set-up

3.1.1 Specify parameters

3.1.2 Sample duplex

```
Number of common edges removed: 79
Number of inactive nodes removed from layer 1: 0
Number of inactive nodes removed from layer 2: 0
Size of active node set union from layers 1 and 2: 200000
```

3.1.3 Compute remnants

3.2 Feature calculation

3.3 Model training and evaluation

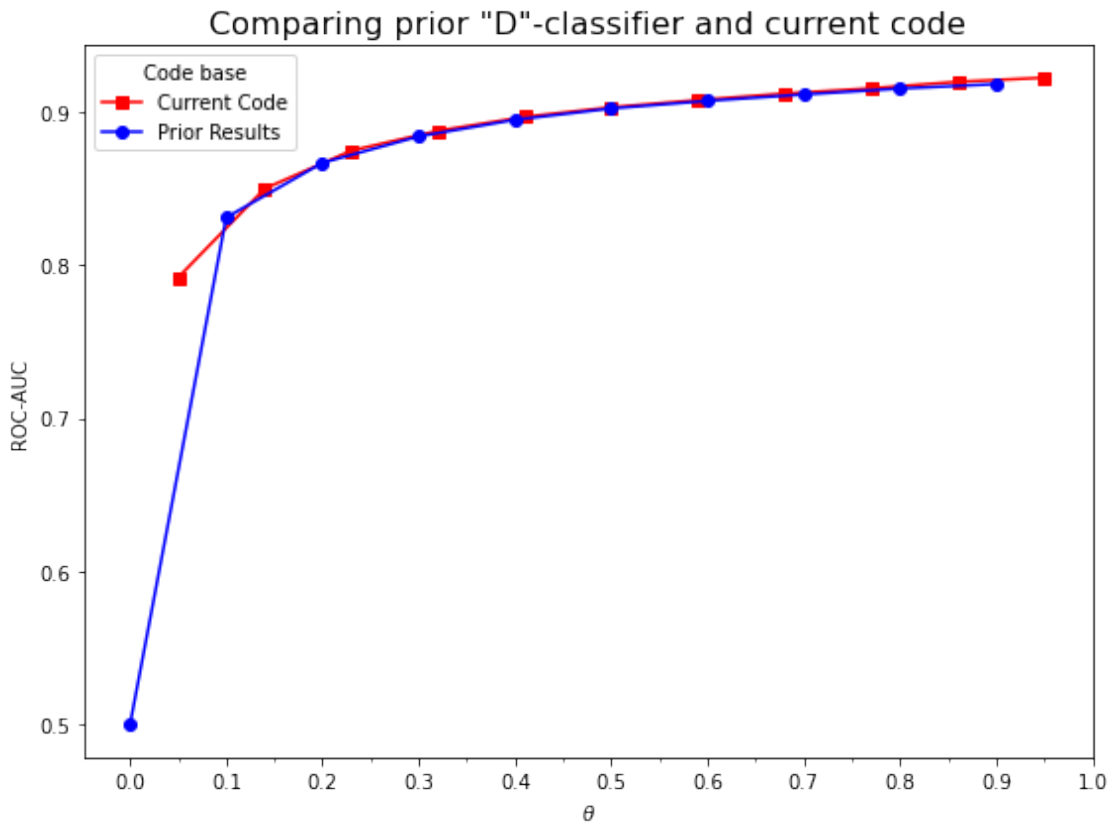
---

4 Analysis

4.1 Retrieving prior results

4.2 Basic performance analysis and comparison

[209]: `Text(0.5, 1.0, 'Comparing prior "D"-classifier and current code')`



The performances appear to match now!

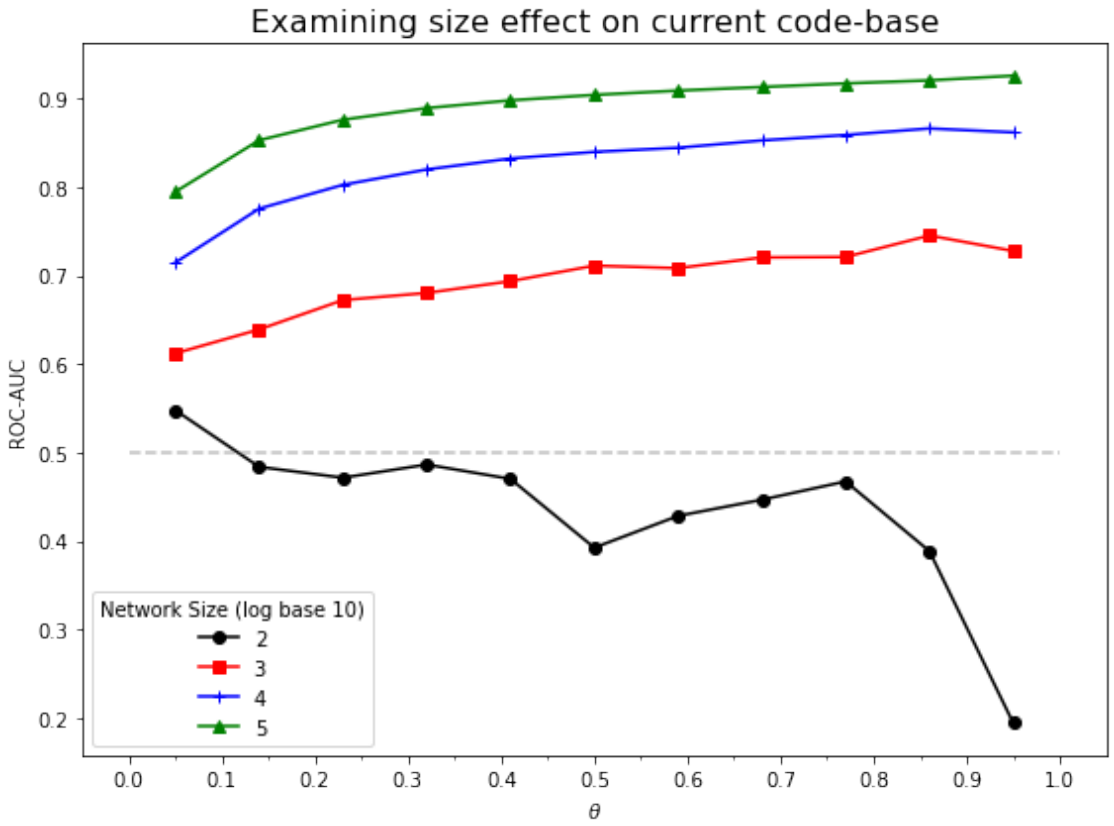
The preliminary results should match the corresponding case by parameters nearly exactly, except for network size. Perhaps the effect of network size is stronger with our current code base than our previous code base?

5 Size Effect

Let's examine our code base by itself controlling network size across a few orders of magnitude.

```
Number of common edges removed: 4
Number of inactive nodes removed from layer 1: 0
Number of inactive nodes removed from layer 2: 0
Size of active node set union from layers 1 and 2: 200
Number of common edges removed: 27
Number of inactive nodes removed from layer 1: 0
Number of inactive nodes removed from layer 2: 0
Size of active node set union from layers 1 and 2: 2000
Number of common edges removed: 58
Number of inactive nodes removed from layer 1: 0
Number of inactive nodes removed from layer 2: 0
Size of active node set union from layers 1 and 2: 20000
Number of common edges removed: 79
Number of inactive nodes removed from layer 1: 0
Number of inactive nodes removed from layer 2: 0
Size of active node set union from layers 1 and 2: 200000
```

[11]: `Text(0.5, 1.0, 'Examining size effect on current code-base')`

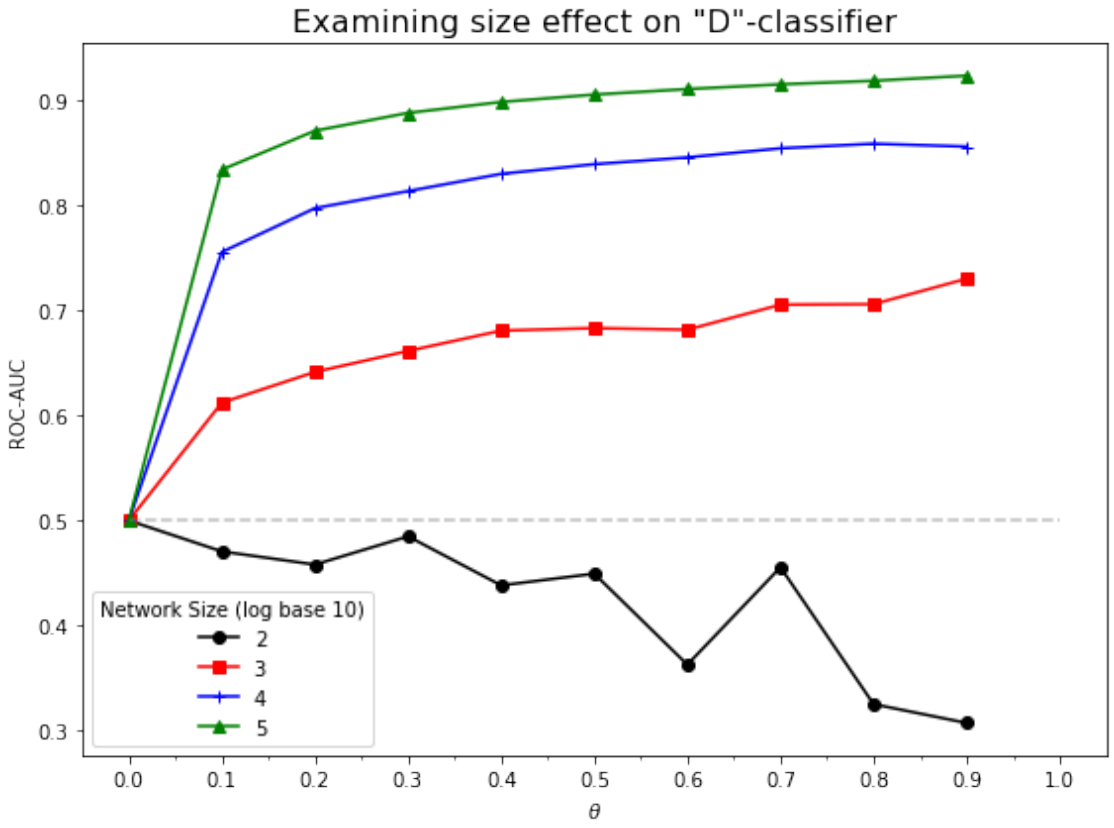


As we can see, there is a pretty strong effect from network size on our current code. I don't recall this being the case for our previous work, so let's recreate the "D"-classifier from the published source code now and verify the size effect is weaker in that setting.

```
Number of common edges removed: 17
Number of inactive nodes removed from layer 1: 0
Number of inactive nodes removed from layer 2: 0
Size of active node set union from layers 1 and 2: 200
# 0.00   0.90
# 0.10   0.90
# 0.20   0.90
# 0.30   0.90
# 0.40   0.90
# 0.50   0.90
# 0.60   0.90
# 0.70   0.90
# 0.80   0.90
# 0.90   0.90
Number of common edges removed: 36
Number of inactive nodes removed from layer 1: 0
Number of inactive nodes removed from layer 2: 0
Size of active node set union from layers 1 and 2: 2000
# 0.00   0.90
# 0.10   0.90
# 0.20   0.90
# 0.30   0.90
# 0.40   0.90
# 0.50   0.90
# 0.60   0.90
# 0.70   0.90
# 0.80   0.90
# 0.90   0.90
Number of common edges removed: 52
Number of inactive nodes removed from layer 1: 0
Number of inactive nodes removed from layer 2: 0
Size of active node set union from layers 1 and 2: 20000
# 0.00   0.90
# 0.10   0.90
# 0.20   0.90
# 0.30   0.90
# 0.40   0.90
# 0.50   0.90
# 0.60   0.90
# 0.70   0.90
# 0.80   0.90
# 0.90   0.90
Number of common edges removed: 75
Number of inactive nodes removed from layer 1: 0
```

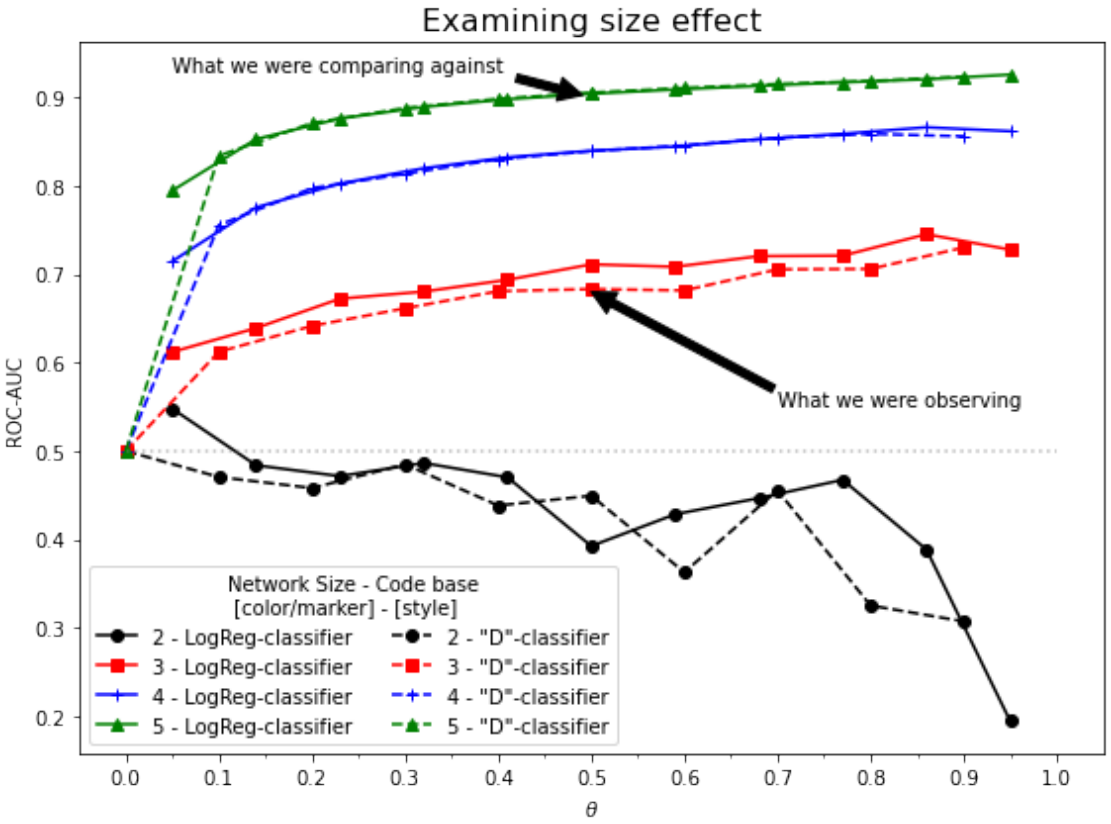
```
Number of inactive nodes removed from layer 2: 0
Size of active node set union from layers 1 and 2: 200000
# 0.00   0.90
# 0.10   0.90
# 0.20   0.90
# 0.30   0.90
# 0.40   0.90
# 0.50   0.90
# 0.60   0.90
# 0.70   0.90
# 0.80   0.90
# 0.90   0.90
```

[15]: `Text(0.5, 1.0, 'Examining size effect on "D"-classifier')`



Ok so it appears I misremembered - there is indeed a seemingly equally strong effect due to network size with the “D”-classifier as well.

Let’s quickly examine the results together



Fantastic! So if we are careful with our network size, the results may very well match up just fine.

6 Appendix

6.1 Referenced figures

Reference Figure (1): EMBex32v2.1 results compared with prior results.