BUSA90501 Machine Learning
Syndicate Project Description

Due date: 9pm Monday 6th July 2020 (competition closes 12pm noon) **Weight:** 30%

1 Overview

Pairwise relationships are prevalent in real life. For example, friendships between people, communication links between computers and pairwise similarity of images. Networks provide a way to represent a group of relationships. The entities in question are represented as network nodes and the pairwise relations as edges.

In real network data, there are often missing edges between nodes. This can be due to a bug or deficiency in the data collection process, a lack of resources to collect all pairwise relations or simply there is uncertainty about those relationships. Analysis performed on incomplete networks with missing edges can bias the final output, e.g., if we want to find the shortest path between two cities in a road network, but we are missing information of major highways between these cities, then no algorithm will able to find this actual shortest path.

Furthermore, we might want to predict if an edge will form between two nodes in the future. For example, in disease transmission networks, if health authorities determine a high likelihood of a transmission edge forming between an infected and uninfected person, then the authorities might wish to vaccinate the uninfected person.

In this way, being able to predict (and correct for) missing edges is an important task.

Your task:

In this project, you will be learning from a training network and trying to predict whether edges exist among test node pairs.

The training network is a fragment of an *academic co-authorship graph*. The nodes in the network—authors—have been given randomly assigned IDs, and an undirected edge between node A and B represents that authors A and B have published a paper together as co-authors. The training network is a network of a time period (2010-2017), focusing on individuals in a specific academic subcommunity.

Your task is to predict if an edge will form between two nodes in the future, we provide development set and test set as future link information to validate and evaluate your works. The development set is a list of 4,866 edges, contain 2,433 real edges in the year after the time period of the training set (2018) , and also 2,433 fake edges (pairs of nodes that are not connected). The test data is a list of 4,460 edges, 2,230 of these test edges are real in the next year after development set (2019), while the other 2,230 do not actually exist.

To make the project fun, we will run it as a Kaggle in-class competition. Your assessment will be partially based on your final ranking in the privately-held competition, partially based on your absolute performance and partially based on your report.

2 Data Format

All data will be available in raw text. The training graph data will be given in a (tab delimited) edge list format, where each row represents a node and its neighbours. For example:

```
1 2
2 1 3 4
3 2 5
4 2 5
5 3 4
```

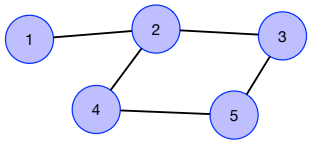

Figure 1: Network diagram for the adjacency list example.

represents the network illustrated in Figure 1.

In addition to the edges, you are also provided with a file including several features of the nodes (authors). This file, "nodes.json" is in JSON format and includes information in 2010-2017 for each author:

- their `id` in the graph
- the number of years since their `first` and `last` publication to 2017 (e.g. `first`:3 means author published first paper at 2014)
- their number of publications in total, `num_papers`
- presence of specific keywords in the titles and abstracts of their publications (denoted `keyword_X` where $X \in \{0,1,\ldots,53\}$, each being a binary value and only listed if its value is 1)
- publication at specific venues (denoted `venue_X` where $X \in \{0,1,\ldots,303\}$, each being a binary value and only listed if its value is 1)

This gives you some additional information beside the network structure for your prediction task.[1]

The test edge set is in a comma separated values (CSV) edge list format, which includes a one line header, followed by a line for each (source node, target node) edge. Your implemented algorithm should take the test CSV file as input and return a 4,461 row CSV file that has a) in the first row, the string "Id,Predicted"; b) in all subsequent rows, a consecutive integer ID, a comma, then a float in the range [0,1]. These floats are your "guesses" or predictions as to whether the corresponding test edge was from the co-authorship network or not. Higher predictions correspond to being more confident that the edge is real.

For example, given the test edge set of $\{(3,5),(4,12)\}$ as represented in CSV format by

```
Id,Source,Sink
1,3,5
2,4,12
```

if your prediction probabilities are 0.1 for edge (3,5), 0.99 for edge (4,12), then your output file should be:

```
Id,Predicted
1,0.1
2,0.99
```

The test set will be used to generate an AUC for your performance; you may submit test predictions multiple times per day (if you wish). During the competition AUC on a 30% subset of the test set will be used to rank you in the **public leaderboard**. We will use the complete test set to determine your **final AUC and ranking**. The split of test set during/after the competition, is used to discourage you from constructing algorithms that overfit on the leaderboard. The training graph "train.txt", the test edges "test-public.csv", and a sample submission file "sample.csv" will be available within the Kaggle competition website. In addition to using the competition testing and to prevent overfitting, we encourage you to generate your own test edge sets from the training graph, and test your algorithms with that.

[1] These features were calculated after excluding from the network the hidden test edges, to invalidate trivial approaches for prediction.

3 Links and Check List

Competition link: `https://www.kaggle.com/t/aedc05f00c12488792c251818b2dd99e`
Team registration: `https://forms.gle/C1KTR6GtEavcXHnb7`

The Kaggle in class competition allows you to compete and benchmark against your peers. Please do the following **by: June 18th 11pm**

1. Setup one account on Kaggle with uni email ending `@student.unimelb.edu.au`.
2. Your project team is your syndicate team.
3. Connect with your team mates on Kaggle as a Kaggle team.[2] **Only submit via the team!**
4. Register your team using the 'team registration' Google Forms link above. **One registration per team.**
5. Complete and upload the 'Group Agreement' form from Canvas, to Canvas to record team-mate expectations within your syndicate.

4 Student Groups

Teams should match assigned **syndicate groups**. We will mark all teams based on our expectations of what a typical syndicate team could achieve: you might consider roles such as researcher, feature engineering, learning, workflows/scripting, experimentation, ensembling of team models, generating validation data, etc. and divide your identified roles among your team. We expect you to complete a 'Group Agreement' found on Canvas with this spec, and upload it to Canvas. We recommend tools such as Slack or Trello for group coordination—you may use your platform of choice.

By the date listed above, please enter the UoM and Kaggle usernames for each team member, along with Kaggle team name—so that we may match teams to students—with the above registration Google Form (one response per team, please).

We encourage active discussion among teams, but please refrain from colluding. Given your marks are partially dependent on your final ranking in the competition, it is in your interest not to collude.

The 'Group Agreement' is important in the process of group work, in setting internal expectations. And platforms like Slack/Trello/Git logs can be used to document contribution (or lack thereof). In the rare circumstance a student is penalised for lack of contribution, that student will have the opportunity to appeal. Again, we don't expect this process to come into effect for any teams—from past experience in a class of this size. In the past students report that this kind of project work is challenging, rewarding and fun.

5 Report

A report describing your approach should be submitted through Canvas **by 9pm July 6th**. It should provide the following sections:

1. A brief description of the problem and introduction of any notation that you adopt in the report.
2. Description of your final approach(s) to link prediction, the motivation and reasoning behind it, and why you think it performed well/not well in the competition.
3. Any other alternatives you considered and why you chose your final approach over these (this may be in the form of empirical evaluation, but it must be to support your reasoning - examples like "method A, got AUC 0.6 and method B, got AUC 0.7, hence we use method B", with no further explanation, will be marked down).

[2] See e.g. `https://www.quora.com/How-do-I-form-a-team-in-Kaggle`

Your description of the algorithm should be clear and concise. You should write it at a level that a postgraduate student can read and understand without difficulty. If you use any existing algorithms, *please do not rewrite the complete description, but provide a summary* that shows your understanding and references to the relevant literature. In the report, we will be interested in seeing evidence of your thought processes and reasoning for choosing one algorithm over another.

Dedicate space to describing the features you used and tried, hyperparameters tuning, any interesting details about software setup or your experimental pipeline, and any problems you encountered and what you learned. In many cases these issues are at least as important as the learning algorithm, if not more important.

Report format rules. The report should be submitted as a PDF, and be no more than five pages, single column. The font size should be 11 or above. If a report is longer than five pages in length, we will only read and assess the report up to page five and ignore further pages. (Don't waste space on cover pages.)

6 Submission

In addition to pre-submission of the 'team registration' Google Form and 'group agreement' PDF to Canvas, the final submission will consist of three parts by the overall project deadline:

- A valid submission to the Kaggle in class competition. This submission must be of the expected format as described above, and produce a place somewhere on the leaderboard. Invalid submissions do not attract marks for the competition portion of grading (see Section 7).
- To Canvas, a zip archive of your source code of your link prediction algorithm in any language including any scripts for automation, and a README.txt describing in just a few lines what files are for (but **no data please**).
- To Canvas, a written research report in PDF format (see Section 5).

The submission link will be visible in Canvas prior to deadline.

7 Assessment

The project will be marked out of 30 and contribute 30 percent towards your subject total mark. **No late submissions accepted. You must inform your lecturer about sickness well before the deadline. Submit early and often to guard against unexpected last minute issues.**

The assessment in this project will be broken down into two components. The following criteria will be considered when allocating marks.

Based on our experimentation with the project task, we expect that all reasonable efforts at the project will achieve a passing grade or higher.

Kaggle Competition (**15/30**):

Your final mark for the Kaggle competition is based on your rank in that competition. Assuming N teams of enrolled students compete, there are no ties and you come in at R place (e.g. first place is 1, last is N) with an AUC of $A \in [0,1]$ then your mark is calculated as

$$12 \times \frac{\max\{\min\{A, 0.80\} - 0.4, 0\}}{0.40} + 3 \times \frac{N-R}{N-1} \ .$$

Ties are handled so that you are not penalised by the tie: tied teams receive the rank of the highest team (as if no team were tied). This expression can result in marks from 0 to 15. For example, if teams A, B, C, D, E came 1st, 4th, 2nd, 2nd, 5th, then the rank-based mark terms (out of 5) for the five teams would be 3, 0.75, 2.25, 2.25, 0.

This complicated-looking expression can result in marks from 0 all the way to 15. We are weighing more towards your absolute AUC than your ranking. **The component out of 12 for AUC gives a score of 0/12 for AUC of 0.4 or lower; 12/12 for AUC of 0.8 or higher; and linearly scales over the interval of AUCs [0.4, 0.8].** We believe that much higher than 0.5 (random classifier) AUC is achievable with minimal work, while 0.8 AUC is an excellent result deserving of full marks. *For example, an AUC of 0.7 for a team coming last would yield 9/15; or 10.5/15 if coming mid-way in the class.*

The rank-based term encourages healthy competition and discourages collusion. The other AUC-based term - rewards teams who don't place in the top but none-the-less achieve good absolute results.

Note that invalid submissions will come last *and* will attract a mark of 0 for this part, so please ensure your output conforms to the specified requirements.

Report (**15/30**):

The below marking rubric outlines the criteria that will be used to mark your report.

Critical Analysis (Maximum = 10 marks)	**Report Clarity and Structure** (Maximum = 5 marks)
10 marks Final approach is well motivated and its advantages/disadvantages clearly discussed; thorough and insightful analysis of why the final approach works/not work for provided training data; insightful discussion and analysis of other approaches and why they were not used	5 marks Very clear and accessible description of all that has been done, a postgraduate student can pick up the report and read with no difficulty.
8 marks Final approach is reasonably motivated and its advantages/disadvantages somewhat discussed; good analysis of why the final approach works/not work for provided training data; some discussion and analysis of other approaches and why they were not used	4 marks Clear description for the most part, with some minor deficiencies/loose ends.
6 marks Final approach is somewhat motivated and its advantages/disadvantages are discussed; limited analysis of why the final approach works/not work for provided training data; limited discussion and analysis of other approaches and why they were not used	3 marks Generally clear description, but there are notable gaps and/or unclear sections.
4 marks Final approach is marginally motivated and its advantages/disadvantages are discussed; little analysis of why the final approach works/not work for provided training data; little or no discussion and analysis of other approaches and why they were not used	2 mark The report is unclear on the whole and the reader has to work hard to discern what has been done.
2 marks Final approach is barely or not motivated and its advantages/disadvantages are not discussed; no analysis of why the final approach works/not work for provided training data; little or no discussion and analysis of other approaches and why they were not used	1 mark The report completely lacks structure, omits all key references and is barely understandable.

Plagiarism policy: You are reminded that all submitted project work in this subject is to be your own individual team work. Automated similarity checking software will be used to compare submissions. It is University policy that academic integrity be enforced. For more details, please see the policy at `http://academichonesty.unimelb.edu.au/policy.html`.