COMP90051 Statistical Machine Learning – 2020 Sem2
Project 1 Specification

Competition link: https://www.kaggle.com/t/98f0d638fa4140e8a3ff86fc3c61d538
Competion closes: 4pm Thu, 17th Sep; **Reports due:** 3pm Fri, 18th Sep. **Total 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 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 partial crawl of the *Twitter social network* from several years ago. The nodes in the network—Twitter users—have been given randomly assigned IDs, and a directed edge from node A to B represents that user A follows B. The training network is a subgraph of the entire network. Starting from several random seed nodes, we proceeded to obtain the friends of the seeds, then their friends' friends, and so on for several iterations.

The test data is a list of 2,000 edges, and your task is to predict if each of those test edges are really edges in the Twitter network or are fake ones. 1,000 of these test edges are real and withheld from the training network, while the other 1,000 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 given in a (tab delimited) edge list format, where each row represents a node and its out neighbours. For example:

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

represents the network illustrated in Figure 1.

The test edge set is in a (tab-delimited) edge list format, where each represents an edge (source node, target node). Given this 2,000-row edge list, your implemented algorithm should take the test list in and return a 2,001

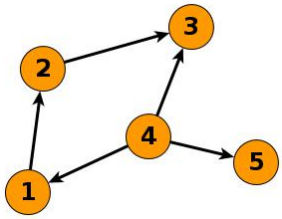

Figure 1: Network diagram for the adjacency list example.

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 Twitter network or not. Higher predictions correspond to being more confident that the edge is real.

For example, given the test edge set of $\{(3,1),(3,4)\}$ as represented in CSV format by

Id,Source,Sink
1,3,1
2,3,4

if your prediction probabilities are 0.1 for edge (3,1), 0.99 for edge (3,4), then your output file should be:

Id,Predicted
1,0.1
2,0.99

The test set will be used to generate an *area under ROC curve* (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 overfitting the test data**—a key concept from class. The training graph "train.txt", the test edges "test-public.txt", 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—a **validation set**—to develop, model select and test your algorithms. Doing so is good practice, and the setup of this project encourages this.

Students should not augment the project data with external data such as further Twitter data.

3 Student Groups

You are expected to work in **groups of 3 students**. We will mark all teams based on our expectations of what a typical 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. As indicated on Canvas ahead of project release, you can either find team mates yourself, we can randomly assign team mates, or both (e.g. 2 with 1).

We encourage active discussion among teams of **conceptual ideas**, but sharing of code, data, or specific solutions or not allowed. Given your marks are partially dependent on your final ranking in the competition, it is in your interest not to collude.

Expected Group Behaviour. We expect all students to contribute equally to their groups, to be communicative, and at all times respectful of fellow students. Your peers will be your future professional networks and possibly co-workers. Group work is an important feature of all professional machine learning workplaces, and a

key generic skill developed by this project. As described below, we require submission of a 'Group Agreement' and at least 3 group meetings with recorded 'Meeting Minutes' during the course of the project. These measures promote positive group dynamics with transparency around group expectations. From past experience of classes of this size, we expect at most a tiny handful of groups to have issues working together.

4 Submissions & Important Deadlines

Submissions in part are due on Kaggle via your Kaggle Team, and on Canvas via your Project Group:

Tue Sep 1, 10AM (optional). If you intend to form your own team in full (3 students), or in part (2 students), you must together join one "Project 1 group" on Canvas. Within a day of this point, we will randomly assign remaining students to groups including completing partial groups of 2.

Fri Sep 4, 6PM. Submit on Canvas your "Group Agreement" through your Canvas group. You are expected to have met with your team members by now; you should together discuss expectations of the group, so that you embark on the project with a shared set of goals and supporting roles. While this agreement is not marked, we require students to complete this.

Thu Sep 17, 4pm. Kaggle competition closes. You must ensure you have made at least one valid submission to the Kaggle in-class competition linked above. Kaggle will allow you to select up to 2 submissions to be used for your scoring.

Fri Sep 18, 3pm. The remainder of your project submission is due:

- PDF final report (see see Section 5 for requirements) submitted through your Canvas group.
- Text of your Kaggle team name only. E.g. if your team name on Kaggle is "TeamStatML" then this submission should only have the text "TeamStatML". If you add additional text, or submit something else, we may not be able to match your Kaggle submission to your team and you could receive zero for your competition performance.
- Zip file of supporting evidence: 3 (or more) meeting minutes following the template on Canvas; code that covers your link prediction algorithm in any language including scripts and automation, and a README. **DO NOT INCLUDE DATA** (to not overload Canvas); any further evidence of team work such as chat/git logs if desired. While we will not mark your code, we may use this information to verify your solution is your own work.

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

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`. You may use software libraries and code snippets found online with attribution, but not code of other University of Melbourne students outside your group.

5 Report Requirements

Your final report 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).

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, 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 three single-column A4 pages. The font size should be 11pt or above. If a report is longer than three pages in length, we will only read and assess the report up to page three and ignore further pages. (Don't waste space on cover pages.)

6 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 lecturers 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 (16/30):

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

$$12 \times \frac{\max\{\min\{A, 0.90\} - 0.4, 0\}}{0.50} + 4 \times \frac{N - R}{N - 1} \ .$$

Ties are handled so that you are not penalised: tied teams receive the rank of the highest team (as if no team were tied). This expression can result in marks from 0 to 16. For example, if teams A, B, C, D, E came 1st, 4th, 2nd, 2nd, 5th, then the rank-based mark terms (out of 3) for the five teams would be 4, 1, 3, 3, 0.

This complicated-looking expression can result in marks from 0 all the way to 16. 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 0.9 or higher; and linearly scales over the interval of AUCs [0.4, 0.9].** We believe that much higher than 0.5 (random classifier) AUC is achievable with minimal work, while 0.9 AUC is an excellent result deserving of full marks. *For example, an AUC of 0.8 for a team coming last would yield 9.6/16; or 11.6/16 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 (14/30):

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

Critical Analysis (Maximum = 9 marks)	**Report Clarity and Structure** (Maximum = 5 marks)
9 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.
7.2 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.
5.4 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.
3.6 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 marks The report is unclear on the whole and the reader has to work hard to discern what has been done.
1.8 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.