COMP90051 Statistical Machine Learning
Project 1 Description

Due date: 5:00pm (competition closes noon 12pm) Friday, 7th September 2018 **Weight:** 25%

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 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

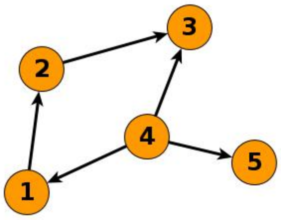

Figure 1: Network diagram for the adjacency list example.

row CSV file that has a) in the first row, the string "Id,Prediction"; 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,Prediction
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.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, and test your algorithms with that.

3 Week 1: Links and Check List

Competition link: `https://www.kaggle.com/t/5a5da7cccc734eb59ea18d0c35630c84`
Team registration: `https://goo.gl/forms/YoGdYFcvWwkaqjad2`
Random pool: `https://goo.gl/forms/JDyukjHqgFtuE0gq2`

The Kaggle in class competition allows you to compete and benchmark against your peers. Please do the following **by 5pm Tue Aug 21**:

1. Setup one (and only one) account on Kaggle with uni email.
2. Form your team of student peers (see below for details). If you need team-mates sign-up yourself (and a friend if desired) with 'random pool' Google Form link above.

Then by the end of the 1st week, **5pm Fri Aug 24**:

3. Connect with your team mates on Kaggle as a Kaggle team. **Only submit via the team!**
4. Register your team using the 'team registration' Google Forms link above.

5. Complete and upload the 'Group Agreement' form from LMS, to LMS to record team-mate expectations within your team.

4 Student Groups

Teams should consist of **three enrolled students**. You can work individually or as pairs, but we encourage you team up in triplets to share the workload. We will mark all teams based on our expectations of what a team of three 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 LMS with this spec, and upload it to LMS. We recommend tools such as Slack or Trello for group coordination—you may use your platform of choice.

If you can't (or don't want to) find partners, please use the sign-up sheet to add to the 'random pool' (you may add a friend with you, if looking for a singleton; only one response for the pair, please), by 5pm Tue Aug 21.

By the end the end of the project's first week, 5pm Fri Aug 24, 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 dependent on your final ranking in the competition, it is in your interest not to collude.

In the event that a group of three loses a team mate due to dropping the subject (as if that would happen, SML is awesome!) please contact the lecturer who will organise an equitable arrangement for the remaining team.

Upon completion of the project we will ask all students to complete a peer self-assessment which is a short, very high-level survey of how you found working with your team-mates. For most teams this will not lead to subsequent action, but for (very few) teams who experienced significant issues we may use these assessments to modify assessment of individual team members. The 'Group Agreement' is important in this process, 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 many (or any!) teams. Almost all teams have lots of fun in project 1.

5 Report

A report describing your approach should be submitted through LMS **by 5pm Sep 7th**. 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).

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 pages, single column. The font size should be 11 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 Submission

In addition to pre-submission of the 'team registration' Google Form and 'group agreement' PDF to LMS (by 5pm Fri Aug 24th please), the final submission will consist of three parts:

- A valid submission to the Kaggle in class competition **by 12pm Fri Sep 7th**. 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 LMS **by 5pm Fri Sep 7th**, 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 LMS **by 5pm Fri Sep 7th**, a written research report in PDF format (see Section 5).

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

7 Assessment

The project will be marked out of 25. Note that there is a hurdle requirement on your combined continuous assessment mark for the subject, of 25/50, of which Project 1 will contribute 25 marks. **Late report submissions will incur a deduction of 2 marks per day—it is not possible to make late competition entries.**

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 (12/25):

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

$$9 \times \frac{\max\{\min\{A, 0.90\} - 0.4, 0\}}{0.50} + 3 \times \frac{N - R}{N - 1} \ .$$

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 12. 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 3, 0.75, 2.25, 2.25, 0.

This complicated-looking expression can result in marks from 0 all the way to 12. We are weighing more towards your absolute AUC than your ranking. **The component out of 9 for AUC gives a score of 0/9 for AUC of 0.4 or lower; 9/9 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 7.2/12; or 8.7/12 if coming mid-way in the class.*

External teams of unregistered students may participate, but their entries will be removed before computing the final rankings and the above expression, and will not affect registered students' grades. We do not actively invite such participation.

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 (**13/25**):

The marking rubric in Appendix A outlines the criteria that will be used to mark your report.

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`.

A Marking scheme for the Report

Critical Analysis (Maximum = 8 marks)	**Report Clarity and Structure** (Maximum = 5 marks)
8 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.
6.4 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.
4.8 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.
3.2 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.
1.6 mark 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.