DA 2180: Probabilistic Machine Learning: Theory and Applications

Jan-April Semester 2024, Project Spec

Due: Competition closes 8th March 2024 11:59PM IST, Report due next day (Saturday) 11:59PM IST

Competition Link: `https://www.kaggle.com/t/2098f01d4c2d442a8b76cd727faec26a`

Weight: 20% of final mark

Introduction

Pairwise relationships are a common occurrence in our daily lives. These relationships manifest in various forms, such as friendships among individuals, communication links connecting computers, and the similarities observed between pairs of cars. One effective way to represent and study these relationships is by employing networks, which are composed of a collection of nodes and edges. In this context, the entities involved are depicted as nodes, while the pairwise connections are denoted as edges.

In real-world datasets, it's not uncommon to encounter instances where certain edges are missing between nodes within a network. This absence of connections can result from various factors, including errors or limitations in the data collection process, constraints on resources preventing the comprehensive gathering of all pairwise relations, or uncertainties surrounding these relationships. Analyzing networks with these missing edges can introduce biases into the final results. For instance, if our objective is to determine the shortest route between two cities in a road network, and we lack information about major highways linking these cities, no algorithm can accurately identify the true shortest path.

Furthermore, there are situations where we seek to predict whether an edge will emerge between two nodes in the future. For instance, in a network tracking the transmission of diseases, if health authorities anticipate a high probability of a transmission link forming between an infected individual and an uninfected person, they may decide to administer a vaccine to the uninfected individual as a precautionary measure. Therefore, the ability to forecast and account for missing edges holds significant importance.

Your Task

In this project, you will be learning from a training network (given in training data) and trying to predict whether edges exist among test node pairs. More specifically, you will be developing/implementing a probabilistic machine learning model to predict the probability of edge between test node pairs. **You can also use non-probabilistic models in this project, however, there will will be** 1 **bonus mark for using any probabilistic ML based innovative approach (given that its use is properly justified in the submitted report (See Report section and rubric).**

The training network is partial crawl of the Twitter social network collected several years ago. In this network, the nodes represent Twitter users and are identified by randomly assigned IDs. A directed edge connecting node A to node B signifies that user A follows user B. It's important to note that the training network is a subset of the entire Twitter network. To create this subset, we initiated the process with a set of randomly chosen seed nodes and then expanded it by including their friends, their friends' friends, and so forth through multiple iterations.

For our testing data, we have a list of 2,000 edges. Your task is to determine whether each of these test edges genuinely exists in the Twitter network or if they are fabricated. Among these 2,000 test edges, 1,000 are authentic edges that have been deliberately kept separate from the training network, while the remaining 1,000 are fake and do not correspond to real connections in the network.

To make the project fun, we will run it as a Kaggle in-class competition. Kaggle is one of the most popular online platforms for predictive modelling and analytics tasks. You will be competing with other students in the class. The

following sections give more details on data format, the use of Kaggle, and marking scheme. Your assessment will be based on your final ranking in the competition, the absolute score that you achieve, and your report. The marking scheme is designed so that you will pass if you put in effort. So fear not and embrace the power of probabilistic machine learning.

Data Format

You will have access to three types of files, primarily train.csv, test.csv, and sample_submission.csv. These files are available on the Class Teams under "Project 1 Discussion" Channel under "Files" section.

The training graph data (train.csv) is given in a adjacency edge list format, where each row represents a node and its out neighbours (users being followed by that node). For example:

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

represents the network illustrated in Figure 1.

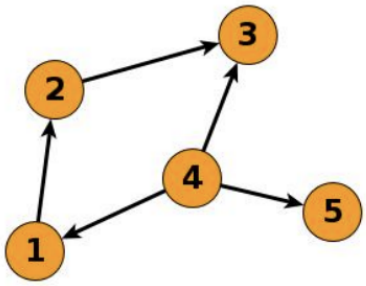

Figure 1: Network diagram for the adjacency list example

Next, the test edge set (test.csv) is also in a edge list format, where each row represents an edge (source node, target node). There are 2000 edges (from 2nd row to 2000th row) in this file. Your task is to predict (in probabilities) whether an edge is true (1) or fake (0).

```
Id    From    To
1     3       4
2     3       1
.....
```

Given this 2,000-row edge list with a unique edge ID (first column) in each row, your implemented algorithm should take the test list in and return a 2,000 row CSV file that has a) in the first row, the header string “Id,Predictions"; b) in all subsequent rows, an ID integer representing row number (1 through 2000) in the first column then a float in the range [0,1] in the second column. These floats are your 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:

A sample_submission.csv file shared with you shows an example submission file. The test.csv and sample_submission.csv comprise same IDs in their first column in the same order. Once you have done predictions for each ID in the test.csv file, you should create a submission file in CSV format. For example, if your prediction probabilities are 0.2 for edge (3,1) and 0.99 for edge (3,4) is true, then your first two rows in submission file will look like this:

Id	Predictions
1	0.2
2	0.99

The test set will be used to generate an AUC for your performance. During the competition, AUC on a subset of the test set will be used to rank you in the leaderboard. We will use the complete test set to determine your final AUC and ranking (using the best of two of your chosen submissions). The split of test set during/after the competition, is used to discourage you from constructing algorithms that overfit on the leaderboard. 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), and test your algorithms with that. This process closely reflects how you would ideally practice machine learning in reality.

Kaggle In-class Competition

Link: `https://www.kaggle.com/t/2098f01d4c2d442a8b76cd727faec26a`
Team Registration Google Form Link: `https://forms.gle/cBzhofHtMq5D3AXv6`
This is a group-based project, so you need to form a team of 3 (max 4 with the permission of the Instructor) students. Please do the following by the **end of the first week** after receiving this project:

- Setup an account on Kaggle with username and email.
- Form your team of student peers (Note that: Some or all teams may be formed by the Course Instructor to make sure each team has student(s) with some prior experience with programming)
- Connect with your team mates on Kaggle as a Kaggle team, using a team name. You can choose any team name e.g., Shaktimaan, Spyderman etc. Only submit your entries via the team; and
- Register your team using the Google Form: `https://forms.gle/cBzhofHtMq5D3AXv6`

Teams should consist of three individuals. If you cannot find a team, please introduce yourself to fellow students in workshop or the lecture, or post to *Project 1 Discussion Channel* in Class Team that you are in search of a team. In only very rare occassions will we permit teams of less than three (and we will mark all teams based on our expectations of what a team of three could achieve). The motivation for working in teams is that in industry, practising machine learning experts work effectively in teams. You should only make submissions using the team name, individual submissions are not allowed and may attract penalties.

The real labels for the test data are hidden from you, but were made available to Kaggle. Each time a submission is made, half of the predictions (50% of the test data) will be used to compute your public score and determine your rank in public leaderboard. This information will become available from the competition page almost immediately. At the same time, the other half of predictions is used to compute a private accuracy and rank in private leaderboard, and this information will be hidden from you. At the end of the competition, only private scores and private ranks will be used for assessment, and will be revealed publicly. This type of scoring is a common practice and was introduced to discourage overfitting to public leaderboard. A good model should generalize and work well on new data, which in this case is represented by the portion of data with the hidden accuracy.

The performance measure we use to evaluate your prediction is the *Area Under Curve* (AUC). The competition server computes an receiver operating characteristics (ROC) curve from your 2,000 probabilities, and the AUC score is the area under this curve. During the course of the competition a public leaderboard will rank teams by the AUC of their latest entry. As stated in the previous section, the leaderboard AUC's are computed on a random subset of the data. The final leaderboard is computed on the entire test set.

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.

Each participant can do maximum 5 submissions everyday. Before the end of the competition, each of you will need to choose your 3 best submissions for final scoring. These do not have to be the latest submissions. Kaggle will compute a private accuracy for the chosen submissions only. The best out of the 3 will then be automatically selected and this private score and the corresponding private leaderboard ranking will be used for marking. If you don't choose any submission, Kaggle will by default consider your best submission performance on public leaderboard for computing the private accuracy.

Report

Each team (only one from the team) will submit a report with the description, analysis, and comparative assessment (where applicable) of the method or methods used. There is no fixed template for the report, but it should provide the following sections:

1. A very 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). Reflect on why the method(s) performed or didn't perform well. If you tried different models or different hyperparameters, compare the methods to each other in the context of this competition. Your reasoning can be in the form of empirical evaluation, but it must be to support your reasoning (examples like "method A with X features and Y value of parameter for accuracy 0.60 and method B, got accuracy 0.7, hence we use method B", with no further explanation, will be marked down).
4. If you used any feature transformations, selected only some useful features, or generated new features, you should also describe them in the report along with the expected effect from using such features and effect observed after implementation and evaluation. In comparing methods, you may want to use an evaluation besides measuring accuracy, in order to better understand the kinds of mistakes being made (e.g., with rare (minor) classes.)

Your description of the algorithms 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, you do not have to rewrite the complete description, but must provide a summary that shows your understanding and references to the relevant literature. In the report, we will be very interested in seeing evidence of your thought processes and reasoning for choosing one algorithm over another. Dedicate space to describing the features you considered and tried, class distributions, validation, sampling techniques (if used), 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.

The report should be submitted as a PDF, and be no more than three A4 pages of content, including all plots, tables and references[1] (single column, font size of 11 or more and margins at least 1 cm, much like this document). You do not need to include a cover page. **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.**

[1]Plots can be useful for model selection, assessing convergence, features importance, displaying results and model interpretation, among other things. For instance, plotting the parameters of your model with respect to the objective function can often give insights into what the model has learned.

Submission and Assessment

In summary, each team (only one student of a team can submit) is required to make the following submissions for this project:

- One or more submission files with predictions for test data (at Kaggle). 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;
- Report in PDF format (via "Assignment" Section for this project in Class Teams);
- Source code used in this project as a single ZIP archive (via "Homework" Section in "Class Notebook" in Class Teams). Your code can be in any of the following languages C, C++, Python, Jupyter Notebook, R or MATLAB. If there is another language you like to use, please contact us first. If the language requires compiling, a makefile or script must be provide to build the executables. We may or may not run your code, but we will definitely read. You should not include the training or test data file in the ZIP file.

Assessments and Marking Scheme

The project will be marked out of 30. No late submission of Kaggle portion will be accepted; late submissions of reports will incur a deduction of 3 marks per day, or part thereof. Based on our experimentation with the project task and the design of the marking scheme below, we expect that all reasonable efforts at the project will achieve a passing grade or higher. So relax and have fun!

Kaggle competition (15 marks) This mark takes into account both achieved accuracy (AUC), as well as your standing in the class. Assuming N is the number of students, and R is your rank in the class, the mark you get for the competition part is

$$12 \times \frac{\max\{\min(acc, 0.96) - 0.50, 0\}}{.46} + 3 \times \frac{N - R}{N - 1}$$

The first term constitutes up to 12 marks, and rewards high AUC system with a maximum score for excellent systems with ≥ 0.96 AUC, and zero score to those with scores $\leq 50\%$ which are as similar as random guessing. The second term, worth 3 marks, is based on your rank and is designed to encourage competition and innovation. Ties are handled so that you are not penalised by the tie. All who are tied will get the same marks for score, but ranking will be decided based on total number of submission entries. The score with less entries will be ranked higher among tied ones.

External teams of unenrolled students (auditing the subject) may participate, but their entries will be removed before computing the final rankings and the above expression, and will not affect registered students' grades. Note that invalid submissions will come last and will attract a mark of 0 for the score, so please ensure your output conforms to the specified requirements, and have at least some kind of valid submission early on!

Report (15 marks) The report will be marked using the rubric in Table 1.

Bonus Mark (1 mark) you will get 1 bonus mark if you have used any ML model which was not taught in the classes/workshops before the submission deadline, or if you have used any innovative techniques in that improves your model performance on test data. You need to provide this information in your report with proper justification, to get this 1 bonus mark.

Critical Analysis (8 marks)	Report Clarity and Structure (7 marks)
7–8 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	*6–7 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
5–6 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	*4–5 marks* Clear description for the most part, with some minor deficiencies/loose ends (e.g., there are no- table gaps and/or unclear sections)
3–4 marks Advantages/disadvantages 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	*2–3 marks* Generally clear description, but there are notable gaps and/or unclear sections.
1–2 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 is unclear on the whole, omits all key reference,and the reader can barely discern what has been done

Table 1: Report marking rubric.

Plagiarism policy

You are reminded that all submitted project work in this subject is to be your own individual work. Automated similarity checking software will be used to compare submissions. It is University policy that cheating by students in any form is not permitted, and that work submitted for assessment purposes must be the independent work of the student(s) concerned. For more details, please see the policy at `https://iisc.ac.in/about/student-corner/academic-integrity/`.