Data Mining Lab 3 – Graphs

INTRODUCTION

Graphs are an important data structure. A significant part of naturally occurring data can be represented using graphs, and the interactions between the nodes in the graph can be used to learn important properties of the underlying data.

In this project you'll represent data using graph structures, analyze the graphs, and implement and run algorithms to help understand the data better and learn from in. Specifically:

- **Spectral embeddings**: Finding low-dimensional representations using eigenvalues and dimensionality reduction techniques.
- **Node embeddings**: Learning to represent graph nodes as meaningful vectors in a common vector space, such that their similarities and other properties are encoded by those vectors.
- **Supervised machine learning**: Using the representations above to predict properties of nodes in the graph.

LEARNING OUTCOMES

After completing this assignment, you will be able to:

- Implement spectral embeddings.
- Implement random walk embeddings.
- Use the node embeddings for supervised machine learning techniques.

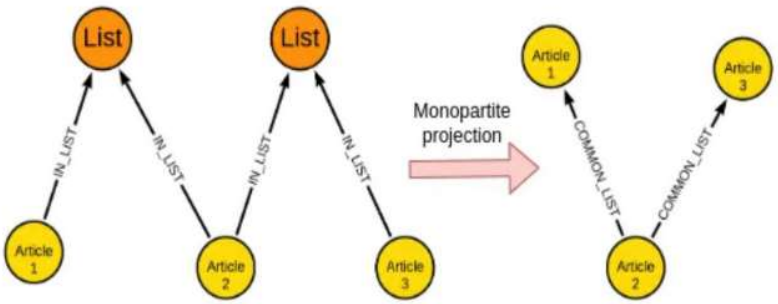
TASKS

The following tasks are to be implemented in WebLabs (see WebLabs for details):

- **Spectral embeddings** (25 points)
- **Random walks** (25 points)

Pipeline assignment:
Medium articles are used to disseminate knowledge and are written ona. Wide range of technical and non-technical topics. Users subscribe to different reading lists where reading lists are either organized as per the domain or certain themes. This naturally gives rise to network structure where articles may belong to similar reading lists and hence are related to each other. Each articles can belong to a certain topic and automatically tagging an article with the topic has immense value for search applications.

A dataset of medium articles along with subscription lists and topic tags. The task is to categorize medium articles to topic tags leveraging the network structure arising from relation using subscription lists. For instance, two nodes are connected if they share atleast one list.



It is a 3-way node classification task

- Forming the adjacency matrix from given raw data after preprocessing (15 points)
- Analyzing graph – number of nodes, number of edges and identify isolated nodes (5 points)
- Get embeddings from Random Walks (15 points)

Analysis:

1. Once the graph is formed, analyze the number of nodes, edges. Since all articles may not belong to common reading lists, there is a possibility some articles are not connected and hence are isolated. Find such isolated nodes and report the count of isolated nodes. You may also need them to handle embedding construction.
2. Compare the performance between node2vec based classification which only relies on network structure and the performance of word2vec+nodevec which considers the semantic features from article titles and network structure. You could plot the embeddings to get an idea of intuition as to why one method works better than other.

Libraries allowed: numpy, pandas, gensim, networkx, sklearn
The submission must be in jupyter notebook format. Use the notebook given and fill in the places indicated by ##START## ##END## by typing in-between the markers mentioned above.

(Kaggle)

The kaggle competition is centered around the task of classifying medium articles to topic tags and is the same dataset as for the pipeline task. The test set labels are held out and training data is provided to the competitors. The competitors are expected to follow a pipeline similar to the pipeline task but are free to use any embedding, random walk mechanism or classifier model to get superior performance. The bonus would be awarded based on improvement over the baseline in leaderboard.

ASSESSMENT CRITERIA

The assignment will be reviewed by your peers, and you are expected to individually review 2 reports. The estimated time you should spend on a review (including code review) is 1 hour. The login details will be provided in the week of the deadline.

Knockout criteria (will not be evaluated if unsatisfied):

Your code needs to execute successfully on computers/laptops of your fellow students (who will assess your work). You may assume the availability of 4GB RAM. Please test your code before submitting. In addition, the flow from data to prediction must be highlighted, e.g., using inline comments.

Submissions submitted after the deadline will not be graded, **deadlines are strict!**

The report/code will be assessed using these criteria:

Criteria	*Description*	*Evaluation*
Spectral clustering	*Test suite score on WebLab.*	*0-25 points*
Node Embeddings	*Test suite score on WebLab.*	*0-25 points*
Embedding-Clustering Pipeline	*Test set accuracy and correctness of pipeline implementation*	*0-20 points*
Pre-processing (graph formation)	*Extracting graph from given medium dataset*	*0-15 points*
Graph analysis		*0-5 points*

Report and code		*0-10 points*
Bonus	*Performance on Kaggle.*	*0-10 points*

Your total score will be determined by summing up the points assigned to the individual criteria. Your report and code will be graded by the teacher and assistants, and the peer reviews are used as guidance.

110 points (including bonus) can be obtained in each lab assignment. 330 points (including bonus) can be obtained in the 3 lab assignments. The total number of obtained points will be divided by 30 to determine the final lab grade.

The lab grade counts for 30% of the total graded for the course.

You will receive a penalty of 10 points for each peer review not performed. Significantly different reviews will be subject to investigation. If deemed badly done by the teacher or TA, you will also receive 10 penalty points.

SUPERVISION AND HELP

We use Mattermost for this assignment. Under channel Questions Lab3, you may ask questions to the teacher, TAs, and fellow students. It is wise to ask for help when encountering start-up problems related to loading the data or getting the expected output from Numpy. Experience teaches that students typically answer within an hour, TAs within a day, and the teacher the next working day. Important questions and issues may lead to discussions in class.

Lab sessions are Friday's 13:45-17:45 physically in different locations at campus (check mytimetable for locations). Please see Brightspace for details.

SUBMISSION AND FEEDBACK

Submit your work in Brightspace, under assignments. Also submit it on peer.tudelft.nl. Within a day after the deadline, you will receive several (typically two) reports to grade for peer review as well as access to the online peer review form. You have 5 days to complete these reviews. You will then receive the anonymous review forms for your groups report and code.

There is the possibility to question the review of your work, up to 3 days after receiving the completed forms. You should do so via the response function on peer.tudelft.nl.

In case of a failing grade for a lab assignment, you have the opportunity to resubmit your work on Brightspace until one week after grade notification.