Network Data Analysis Coursework Instructions

The coursework consists of two parts (Part 1, which covers weeks 1-5; and Part 2, which covers weeks 6-10), and each part contains a series of connected tasks. In each task, you should consider and apply the ideas and techniques learnt in the module along with your technical skills and analytical mindset. The tasks are not a set of options: **all tasks must be completed by each student**. The submission will be a single written report, covering both parts and all tasks. All your code must be committed and pushed to a repository accessible by markers, and a link to this repository must be included in your report.

Please read this full set of instructions, which contain:

- For each Part 1 and Part 2
    - Primary Data
    - Tasks
- Assessment
- Submission instructions

Part 1

Primary Data

The data files for Part 1 are available at the following link (also here):

https://emckclac-my.sharepoint.com/:u:/g/personal/k2037030_kcl_ac_uk/EZ8lHH3PWrVKuNJcG0Tq84IBvy9YHvsQ9VPuUCoQCToqBA?e=PeY0Uc

The data represents the social network of editors that voluntarily contribute to Wikidata [1], a collaborative, global, and cross-domain knowledge graph. Wikidata is conceptually and technologically similar to Wikipedia, the global encyclopedia anyone can contribute to; the main difference is that instead of text articles as in Wikipedia, Wikidata is a structured database about entities in the world.

Just as in Wikipedia, Wikidata editors interact as a community in various ways, both contributing edits (adding, removing, editing content) as well as discussing and talking between them. For this project, we will focus on the latter and study their interactions through **Talk pages**. Talk pages are special pages in which editors can talk among each other about a variety of subjects: discuss issues around a particular item or property (ITEMS, PROPERTIES), talk to other users in their unique pages (USERS), vote for or against in the proposal of new properties (PROPERTY_PROPOSAL), and a large et cetera.

Talk pages have a page name, a number of threads, and under each threat, users can add their comments. Any user can open a new thread or make a comment on an existing one on any page. Pages are always related to a specific item, user, issue, etc. they discuss about; threads are used to separate subjects or different topics to organise the discussion; and users always sign their comments with their username to know who said what.

This project starts with a CSV file with three columns: **page name**, **thread subject**, and **user name**. Each row indicates that the user with the given username made a comment to a particular thread in a specific page. This creates a social network of communication that connects Wikidata users together when they have direct conversations and discussions in these Talk pages.

You can consider extending the dataset with additional information (geographic, demographic, etc.) at your own discretion, if that can improve the depth of your analysis.

Of the full collection of data files, **your must choose 3 for further analysis**. Of those 3, **one must contain a network of large size, another of medium size, and another of small size**. You can use the file size as a proxy for network size. You can ignore the rest of the files, as each of these contain independent networks for the various talk pages.

[1] https://www.wikidata.org/wiki/Wikidata:Main_Page

Tasks in Part 1

You must complete Task A, B and C for each of the 3 selected graphs. Task D requires that you have completed the previous tasks as it is about comparing these networks among them.

Task A (network construction):

You should first turn this data into a *Wikidata editor network* for analysis. For this assignment, you should assume that Wikidata user editors are represented by nodes in a network, and their social connections represented by edges. **Two users have a social connection iff they both have made a comment in the same thread and in the same page**. Try to make your code generic enough as to extract different networks according to different definitions.

Turning the data into a Wikidata editor network will require thinking through the coding problem (i.e. assigning IDs to nodes, bookkeeping user names and their corresponding node IDs, etc.) and using libraries such as pandas and networkx. You might find it helpful to look at the "Converting to and from other data formats" section of the networkx online documentation.

You should then answer the following questions:

- What elements of the dataset you represented as nodes and edges?
- What data structures did you choose to represent this network? How do you keep additional information that cannot be directly encoded as nodes and edges?
- What was your algorithmic approach for building the network?

The documentation of this task should be outputs from running code that answers these questions, visualisations as appropriate, and accompanying textual explanations for each of these, i.e. as you might see in a well documented Jupyter Notebook or like the code example pages provided on KEATS for this module. You do not need to include every decision you followed, just those showing the most important and interesting steps.

Task B (network metrics)

One you have built the network, you should analyse it using the analytical tools, metrics and algorithms that we have seen in class. Your perspective should be in situating the Wikidata editor network within the regular network - small world network - random network spectrum.

To do that, you should answer the following questions:

- What are the characteristic properties and relevant metrics and distributions of this Wikidata editor network?
- How different is this network from a random network, according to the various comparison criteria we have seen in class?
- If you take this network as a complete and representative description of editor networks in collaborative knowledge projects like Wikipedia and Wikidata, what does it tell you about the way editors talk among themselves and their social activity and connections?

The documentation of this task should be outputs from running code that answers these questions, visualisations as appropriate, and accompanying textual explanations for each of these, i.e. as you might see in a well documented Jupyter Notebook or like the code example pages provided on KEATS for this module. You do not need to include every decision you followed, just those showing the most important and interesting steps.

Task C (epidemic models):

Consider a situation in which the Wikimedia Foundation (the entity governing Wikimedia projects, including Wikidata and Wikipedia) would like to track “trolls” and know about how opinions and controversial topics expand and propagate among editors. When an editor makes lots of comments in the same page and thread and many other users reply, there is a chance these are about some controversial topic that will propagate to other talk pages and editors. The Foundation would like to monitor for signs of this propagation and see if they can quickly predict what aspects are more controversial for editors. To do so, two randomly chosen editors are selected every day and checked whether users have made comments to the same page and thread more frequently than usual. The Foundation wants to answer questions such as:

- If both editors have been commenting more than usual on the same day (i.e. might be controversial, trolling, etc.), how can they use the network data to judge how plausible it is that this behaviour has not propagated yet to neighbouring similar editors?
- If one or both of the tested editors have indeed been “possibly trolling”, then the Foundation will start checking other editors as well, prioritising those with a higher chance of having been trolling. How should they use the data to come up with a priority list on what editors to check first?

To think about the above, you could consider shortest paths in networks, numbers of similar editors, cascading, epidemic models etc. For the second question, the situation might be quite different depending on whether one editor has been trolling, or both were. The documentation of this task should be first, a textual explanation of how you would tackle the issue; and, second, outputs from running your code along with explanations and examples.

Task D (comparing networks and social issues):

Consider the following two facts:

- Despite all belonging to the same community, the 3 different graphs you have selected above may yield different results. They may contain different topologies sitting at various places in the regular - small world - random network spectrum
- Despite the definition we have used to connect two editors, there is a subjective and social aspect to decide when and how to establish an edge between the nodes of two editors. This can make obvious links in the network not so obvious (and less prone to propagation) and thus have effects in the analysis. We want to use network metrics on the Wikidata editor network to assess the quality of its links.

According to the above, consider the following questions:

- How do the 3 networks that you have chosen compare in terms of the results of the previous tasks? Where do these 3 networks sit in the regular, small world, and random network continuum? What metrics and results tell you that this might be the case, and how confident can you be about them?
- What kind of problems can impact the quality of the network? How would you address them?

This task does not (necessarily) involve any coding. The documentation of this task should be a textual description discussing the characterisation of your 3 networks according to your previous results, and explaining their similarities/differences and their location in the regular/small world/random network continuum; and suggesting how the choices you have made for representing networks may have an impact in your results.

Part 2

Primary Data

Part 2 asks you to analyse the road network and road events in the centre of a UK city, Leeds. The road network of Leeds is available from OpenStreetMap, while datasets on road traffic accidents in Leeds over a few years are available from the following source:

https://data.gov.uk/dataset/6efe5505-941f-45bf-b576-4c1e09b579a1/road-traffic-accidents

Part 2 consists of a series of connected tasks. In each task, you should consider and apply the ideas and techniques learnt in the module along with your technical skills.

Tasks in Part 2

Task A (spatial networks and planarity):

In this task, you construct and investigate the road network. You are asked to choose an area of approximately 1 square kilometre around the centre of Leeds for your analysis. You should look for an area where a significant number of the recorded road accidents occurred in the area, e.g. 300 or more total across multiple years, and show you have tested this in your report. You only need to consider roads used for driving, not walking paths or private roads (investigate the osmnx documentation for how you might do this filtering). You should then answer the following questions:

1. What are the coordinates of the chosen area?
2. What are the characteristics of this road network? Include, at least, the spatial diameter of the network, the average street length, node density, intersection density, and edge density.
3. What is the average circuitry of the network? What does this tell you about the efficiency of using roads in this area?
4. Is the network planar? Why/why not? Provide examples and argue your answer considering the conditions of planarity.

Task B (road accidents):

In this task, you investigate the road accidents on the road network.

1. Plot the distribution of road accidents on your road network and visualise this. Aggregate across multiple years of accident data. You do not need to consider or represent when the accidents happened, only their location; but you are welcome to add information about time if you believe there is something interesting to show.
2. Investigate whether a high number of accidents on one road correlates with a high number on connecting roads. Calculate the k-function and the Moran's I values for the above spatial graph. What inferences can you draw from this analysis?
3. Investigate whether accidents happen nearer to intersections or partway along roads. Consider this as asking at what fraction of the road length away from the nearest intersection do accidents typically occur.

For the above, we suggest using the spaghetti library as shown in your lessons. The third question will require investigating the API of the library to find the relevant functions to answer the question.

Task C (Voronoi diagrams):

Despite its accidents, the city of Leeds is ideal for organising marathons. The city mayor would like to organise a day of parallel, simultaneous marathons in different parts of the city. The mayor would like to maximise the participation of citizens by organising these marathons in diverse locations of the city, dividing the city into various areas (or “cells”) so that every person can join a marathon that is close to their home. Within each of these cells, a path of exactly 42 Km is needed. Assuming that the mayor would like to organise N=4 simultaneous marathons:

1. Select the initial set of 4 cell seed points. For this, you can use several criteria, such as being far away from frequent accident roads, being close to public transport, being evenly spread, etc. (explain your choice in the report).
2. Visualise the cells yield by your selection of seed points in a Voronoi diagram. What kind of Voronoi diagram (edge planar, node network, or edge points network) is most useful for this problem, and why?
3. Find 2 or 3 cells for which you can find at least one path (or more, if possible) that is (a) approximately 42 Km long, and (b) finishes at the same point where it starts. Visualise both the cells and the found paths.
4. Try to extend the previous step to all cells. Can you find at least one such a path for every cell?
5. If for steps 3-4 there were cells with no such path, what different options could you consider to increase the number of cells that include such paths? (Hint: think about

the number and location of seed points; the size of the area under consideration; etc.) Choose one of such options, repeat steps 3-4, and report the results you obtain, explaining your reasoning.

Task D (TransE, PROV, PageRank):

The mayor's office is also interested in finding an efficient way of representing the provenance of important events in the road network of Leeds and how this could be used for insights.

1. How would you represent marathons using the W3C PROV provenance data model standard[1]? What would correspond to Agents, Entities, and Activities? Provide a diagram illustrating your modelling, and create a (not necessarily spatial) network with (not necessarily real) data, with at least 20 nodes in 1 single connected component.
2. Compute the PageRank value for all the nodes of this network. What do these PR values tell you about the events in the city?
3. Train and evaluate TransE, RotatE and GCN embeddings for with the CoDExMedium dataset, visualise them, and evaluate them on the previous provenance network. Tune hyper-parameters to improve performance. What could these embeddings be used for from a practical point of view? What kind of problems could they help address?

Assessment

The following criteria will be used to determine the mark for each submission:

- Demonstrated understanding of network data analysis concepts and how they can apply to the questions in the coursework tasks.
- Technical ability in using programming to tackle a data analytics problem, showing ability to research and apply data manipulation techniques as required for the problem.
- Creative thinking about the problems described in the coursework and specifically the network-related aspects.
- Clarity of explanation of what code does and why and what results mean, plus good use of visualisations and presentation.
- Succinctness of reporting, i.e. conveying a lot of substance clearly in a short amount of space. Note that marks will be deducted for exceeding the page limit.
- Ability to interpret, explain and position research papers related to the tasks in the coursework.

Submission instructions

Deadline. The strict deadline is **4pm UK time on 10 April 2025**.

Size limits. The report should be at most 16 pages in length including figures, but excluding any references, for which you have unlimited space, and excluding the cover page. Use the cover page to include your name, k-number, and link to your code repository. You are

[1] https://www.w3.org/TR/prov-primer/

welcome to add an appendix beyond the page limit if you want to document more work you have done. The amount of space used per task may vary depending on how much you find to say on each.

Submission format. The report should be submitted on KEATS as a single PDF document. You must include a link to the code repository where you have deposited your code in the cover page (also include your name and k-number).

Plagiarism, collusion and technical support. You are not allowed to submit anyone else's work as your own (plagiarism), which is a serious matter of misconduct. Do not copy text from other papers without appropriate academic practice (citation and reference to the paper **and** quotation marks for all text copied). Do not copy text from ChatGPT or other text generators. You and you alone are entirely responsible for the text contained in your report.

You are allowed and encouraged to discuss specific technical problems you face with the coursework and how to solve them with the rest of the class via the KEATS discussion forum. For full guidance on what is acceptable to ask the class and what must be done individually see the 'Coursework questions and collusion' page in the Assessment section of the KEATS page and feel free to ask clarifying questions in the tutorials or the discussion forum.