CSE891 Project Part 1: Data Creation and Exploration

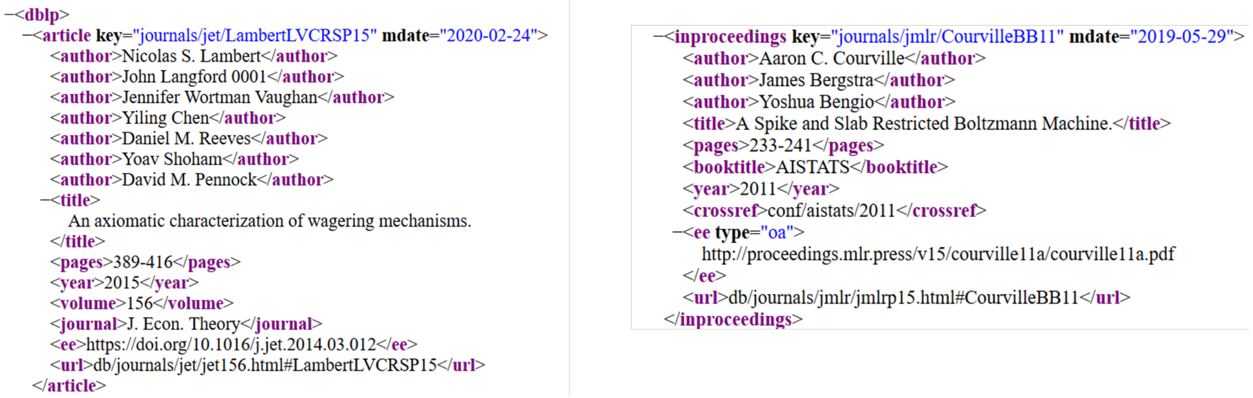
This mini project accounts for 10% of your final grade. The project must be done individually, without collaboration with other students. You are required to implement and apply machine learning methods for analyzing social network data. Your code must be implemented in Python on Jupyter notebook so that each step can be verified for correctness. You are prohibited from using any Python libraries for network analysis or online source code from other authors for this project. You are expected to complete the project using numpy and the standard functions in Python. Please check with the instructor first if you want to use other packages besides those provided by numpy and other packages described below. The project due date is Sunday, Dec 13, 2020 (before midnight). This is the first part of the project description, which accounts for 50% of the project grade. You are strongly encouraged to start the project early.

1 Project Overview

The goal of this project is to provide you with hands-on experience applying machine learning methods to real-world data. The project involves all aspects of the data analysis pipeline—from dataset creation to data exploration and analysis. For this project, you will use a sample dataset from DBLP, a bibliography database of computer science publications (including journals, conference proceedings, book chapters, technical reports, etc). Your goal is to extract a network of co-authorship relation between researchers identified to be among the top researchers in machine learning (i.e., those in the top-300 highest h-index according to Google scholar and have collaborated at least once with each other). The list of these authors names and their corresponding DBLP entries will be provided as input data for your code.

The specific tasks to be performed on this project are as follows:

1. **Network data creation**. In this step, you goal is to create a co-authorship network from the given input files and store their relation in an output ASCII text file named `coauthor.csv`, which contains 2 comma-separated columns, (authorID1, authorID2), if the two authors have collaborated at least once in the past. The mapping from author name to author ID is given in the file `authors.txt`. For this step, you may use python's XML minidom library to load and process the input file. Your code should focus on extracting the co-authorship relation from journal and conference proceedings collaborations only. Other types of collaboration (as editors or authors of books, book chapters, technical reports, etc. should be ignored). You can identify whether a publication corresponds to a journal article or a conference proceeding by their `<article>` and `<inproceedings>` XML tags (see Figure 1). You only need to extract co-author relationship between authors whose names appear in the `authors.txt` file (you can ignore the relationship involving other authors who are not on the list). For example, in Figure 1(right), only 2 of the 3 authors, Aaron C. Courville and Yoshua Bengio, are in the `authors.txt` file. So, you should only output a co-authorship relation between the two of them and ignore James Bergstra. Observe that some authors have trailing digits in their names (e.g., John Langford 0001 or Fei-Fei Li 0001) to disambiguate multiple authors with the same name. You should include the trailing digits when searching the DBLP entries for those authors. Finally, note that the DBLP dataset is noisy, i.e., contains errors. For example, some author names may appear twice in the same article. You may ignore the duplicate occurrences of the same author in the same article. In addition, some authors



```
–<dblp>
  –<article key="journals/jet/LambertLVCRSP15" mdate="2020-02-24">
      <author>Nicolas S. Lambert</author>
      <author>John Langford 0001</author>
      <author>Jennifer Wortman Vaughan</author>
      <author>Yiling Chen</author>
      <author>Daniel M. Reeves</author>
      <author>Yoav Shoham</author>
      <author>David M. Pennock</author>
    –<title>
        An axiomatic characterization of wagering mechanisms.
      </title>
      <pages>389-416</pages>
      <year>2015</year>
      <volume>156</volume>
      <journal>J. Econ. Theory</journal>
      <ee>https://doi.org/10.1016/j.jet.2014.03.012</ee>
      <url>db/journals/jet/jet156.html#LambertLVCRSP15</url>
    </article>
```

```
  –<inproceedings key="journals/jmlr/CourvilleBB11" mdate="2019-05-29">
      <author>Aaron C. Courville</author>
      <author>James Bergstra</author>
      <author>Yoshua Bengio</author>
      <title>A Spike and Slab Restricted Boltzmann Machine.</title>
      <pages>233-241</pages>
      <booktitle>AISTATS</booktitle>
      <year>2011</year>
      <crossref>conf/aistats/2011</crossref>
    –<ee type="oa">
        http://proceedings.mlr.press/v15/courville11a/courville11a.pdf
      </ee>
      <url>db/journals/jmlr/jmlrp15.html#CourvilleBB11</url>
  </inproceedings>
```

Figure 1: Sample DBLP entry in XML format.

can have different variations to their names (e.g., Alex Smola and Alexander Smola or Tom Mitchell and Tom M. Mitchell). You should ignore the name variations and use only the name provided in the `authors.txt` file. You should also be aware of Unicode characters in some of the author names when doing string comparison. Finally, note that the DBLP entries are stored in multiple files: `dblp1.xml`, `dblp2.xml`, $\cdots$, `dblp9.xml`. You can test your code to make sure it works on 1 DBLP file first, but must then apply it to all the files when creating the co-authorship relation. Your code should include a loop to iterate through all the 9 input files.

2. **Data exploration**. For this step, you need to load the `coauthor.csv` data created in step 1 and store it into a 2-dimensional array called adjmatrix. The array represents a 2-dimensional adjacency matrix of the co-authorship network, whose entries are either 0 (if the authors have not collaborated in the past) or 1 (if the authors have collaborated in the past). Since the dataset is only a small sample of the entire DBLP authors, you should be able to store it as a regular (2-d array) without the need to use sparse arrays. Check to make sure that the adjacency matrix is symmetric to prevent errors in the remaining part of the project. After loading the data, you must perform the following data exploration steps:

   (a) Compute the number of nodes (vertices) and links (edges) of the network. Since the graph is symmetric, the number of distinct co-authorship relation should be half of the total number of links.

   (b) In network analysis, centrality is is a measure of the importance of the nodes in a graph. There are many ways to determine the centrality of a node. For this project, you need to compute the following two centrality measures:

      i. **Degree Centrality**. Compute the degree (i.e., number of collaborators) associated with each author. Identify the names (not just ID) of the top-5 researchers with highest degree centrality. Plot a histogram to display the resulting degree centrality of the nodes.

      ii. **Eigenvalue Centrality**. Find the principal eigenvector of the adjacency matrix using power method. Identify the names (not just ID) of the top-5 researchers with highest absolute value in the principal eigenvector. Plot a histogram to display the resulting eigenvalue centrality of the nodes.

(c) Compute the local clustering coefficient of each node in the graph. Given a node w, its local neighborhood, $N(w)$ is defined as all the neighboring nodes adjacent (connected) to w. The local clustering coefficient for w is defined as follows:

$$C(w) = \frac{\sum_{u \in N(w)} \sum_{v \in N(w)} A(u,v)}{d_w(d_w - 1)},$$

where d_w is the degree of w and $A(u,v)$ is equal to 1 if u and v is connected. For example, if node p has 3 neighbors, $\{q,r,s\}$, and there are links between (q,r) and (r,s) but none between (q,s), then the clustering coefficient for p is $C(p) = 4/6$. If a node has only 1 neighbor, you can assume its local clustering coefficient is 0. Identify the names (not just ID) of the top-5 researchers with highest clustering coefficients. Plot a histogram to display the clustering coefficients of the nodes.

3. **Data analysis**. Details will be given in part 2 of the documentation.

Deliverables: Submit your Jupyter notebook (`part1.ipynb`) along with its HTML version to D2L.