Lab 7: PageRank and Link Analysis

Due date: Friday, June 1, midnight.

Note: Submissions will be accepted without penalty until Thursday, June 7, 10:00am (end of the final exam time).

Overview

In this assignment you will implement PageRank ranking algorithm and run it on a number of datasets.

Assignment Preparation

This is a pair programming assignment. Each student teams up with a partner. Each team submits only one copy of the assignment deliverables.

Data

There is a number of datasets available for this assignment:

1. **STATES.** This dataset contains information about the 48 mainland U.S. states plus the District of Columbia and the borders that these states share. (Alaska and Hawaii are excluded from the dataset since they do not share borders with other states).
2. **NCAA-FOOTBALL.** This dataset contains information about **every single game** played by Division I teams in the 2009 NCAA regular football season (before bowls and championships started). A total of 1537 games was played, their results are documented in the dataset.
3. **KARATE.** This dataset describes a small social network consisting of members of a university karate club.

4. **DOLPHINS.** A social network of a group of dolphins as observed by researchers over a period of time.

5. **LES-MISERABLES.** A graph of co-occurrence of difference characters in the chapters of Victor Hugo's novel *Le Miserables*.

6. **POLITICAL-BLOGS.** A graph of political blogs connected by their citations of each other on the eve of the 2004 Presidential election in the US.

General Data Format. To simplify parsing, all datasets are available to you in a uniform data format. This format makes most sense for the football games, but has some redundant information for other datasets.

The data is presented in a CSV (comma-separated values) format. All datasets but **NCAA-FOOTBALL** have four columns in the CSV file. **NCAA-FOOTBALL** adds a fifth column, which is optional for processing (it is only useful if you go for extra credit, and even then you can ignore it).

The generic data format is:

```
<Node1>, <Node1-Value>, <Node2>, <Node2-Value>
```

Each row in the format above represents a single edge in the graph. Here, `<Node1>` and `<Node2>` are **unique ids** of two nodes forming an edge. `<Node1-Value>` and `<Node2-Value>` are two numeric labels that can be associated with an edge. In most of the datasets, one, or both of these values are set to 0. If non-zero values are present, they can be used to determine the direction of a link (if the dataset represents a directed graph).

Specific instructions regarding the data format for each of the datasets are included below.

STATES. The dataset represents an undirected graph. One .csv file, `stateborders.csv` is provided. Each edge in the graph is repeated twice. The node values in each row are always 0: e.g., the border between Florida and Alabama is represented by the following two rows in the CSV file:

```
"AL",0,"FL",0
"FL",0,"AL",0
```

NCAA-FOOTBALL. The data format includes a fifth, optional column and is interpreted as follows:

```
<Team1>, <Team1_Score>, <Team2>, <Team2_Score>, <overtime>
```

Here, the node labels are the names of the NCAA football teams, and the node values are the number of points each team scored in a game. The

graph is directed. For each game played, an edge starts at the team that lost the game and ends at the team that won it. The winning team is listed first in the row (it is `<Team1>`), but this is not a given - to determine direction of an edge you need to get the score. The fifth, optional column indicates if the game involved an overtime.

There is only one data file, `NCAA_football.csv`. A sample of entries from the file is below:

```
"Ball State", 48, "Northeastern ", 14,
"Central Michigan", 31, "Eastern Illinois", 12,
"Southeast Missouri State", 35, "Southwest Baptist", 28, (OT)
```

Due to conversion issues, some scores are stored as integers, and some — as strings (e.g., `" 17"`). Your parser should strip the second and the fourth column value of everything except for digits.

KARATE. The dataset represents an **undirected graph** of social interactions. (If X interacted with Y, then Y interacted with X). There are two CSV files: `karate.csv` and `karateDir.csv`. The former represents each interaction as two rows in the CSV file in order to maintain symmetry. The latter represents each interaction using a single row. The node values are always 0. The node labels are integers from 1 to 34.

DOLPHINS. Same type of graph, data format and file availability (`dolphincs.csv` and `dolphinsDir.csv`) as for the **KARATE** dataset. The only difference is that node labels (names of individual dolphins) are strings enclosed in double quotes.

LES-MISERABLES. The dataset represents an **undirected, edge-labelled** graph of co-occurrences of charactes in the novel. Node labels are strings representing character names. The first node value is the edge label, representing *the total number of chapters in the book, where the two characters co-occur*. The second node value is always 0. Two CSV files are available, `lesmis.csv` and `lesmisDir.csv`. The former file represents each co-occurrence with a pair of entries, e.g.:

```
"Napoleon",1,"Myriel",0
"Myriel",1,"Napoleon",0
```

The latter file uses only one of the two rows.

POLITICAL-BLOGS. The dataset represents a **directed graph** of citations among political blogs. Citations are modeled as **incoming edges** in the graph. There is one data file available, `polblogs.csv`. Node labels are integer ids. The first node label is the citing blog, the second node label

is the blog *being cited*. The first node value is always 0, the second is always 1, encoding the fact that each row represents an outgoing edge for the first node and an incoming edge for the second.

GML format. Additionally, all datasets but `STATES` and `NCAA-FOOTBALL` are available in GML (Graph Markup Language) format. GML format allows for some flexibility in representing graphs (a variety of means can be used to encode meta-data about the graph, and node and edge labels). GML representations may contain extra information about the graphs (e.g., the GML file of the **POLITICAL-BLOGS** dataset contains a "liberal" or "conservative" label for each blog and identifies it). You are welcome, but are not required, to use the GML files provided to you.

All data is found on the Lab data page:

http://users.csc.calpoly.edu/∼dekhtyar/466-Spring2012/labs/lab07.html

Lab Assignment

Write a program that takes as input a data file formatted in the way described above, runs the PageRank analysis on the graph extracted from the input file and outputs the individual items (football teams, states, etc...) ranked in descending order of their computed PageRank together with the PageRank score and the rank.

The program, `pageRank.java` shall take as input the file name. It may also (if you want) take as input a flag specifying whether the dataset you are reading in represents a directed or undirected graph (some of you may find it useful, but it is not absolutely necessary so it is left up to you. Please specify in the README file if you have the flag and if it is mandatory for your implementation).

It should parse the input, create a graph structure from it in main memory and run a version of PageRank ranking algorithm to rank the nodes in the graph.

You may implement the version of PageRank that was discussed in the class. You may also use extra information available to you (in case of the football season data: the score of the game and whether there was an overtime; in case of the *Les Miserables* data, the number of chapters/"thickness" of each connection) to adapt your PageRank computation.

Your program should output an ordered list of **all** nodes in the graph. It should print the rank and the PageRank score of each of them. For example, your program can output:

```
1       obj: Mississippi with pagerank: 0.030110446720353286
2  obj: Florida with pagerank: 0.02406493620983336
3  obj: Utah with pagerank: 0.01609201202237497
4  obj: Oklahoma with pagerank: 0.01531666186988849
```

as the first four items for the `NCAA_Football.csv` input file.

(note, the actual results may vary from the one above)

Report

The assignment will be graded primarily based on the contents of your report. Your report shall be a word-processed document submitted, preferably, in PDF format, which contains the following information:

1. **Front matter.** Title, course number, lab number, names of team members.
2. **Implementation Overview.** A short text (a few paragraphs) describing the details of implementation of your version of the PageRank algorithm. If you made any changes, or added any options to the algorithm, describe them as well.
3. **Results.** For each dataset, include a subsection which contains the following information:
   (a) Any specific information about the settings (if any) used to run PageRank on this dataset.
   (b) A concise copy of the output produced by your program on the dataset. *Concise* means that if your dataset is large (the output is more than two pages long), you can trim the output to show only the top — the most important, according to your algorithm — items from the dataset. The output should be typeset in a contrasting font from the rest of the report (e.g., use a `typewriter-style evenly spaced font` for the output).
   (c) Any observations, comparisons you can make about the work of PageRank on the dataset. Essentially, I am interested in your opinion on whether PageRank really did discover the proper ranking of the items in the dataset.
4. **Overall summary.** Provide a short overall summary of the observed results. How did PageRank work? Did it produce good rankings for most of the datasets? What types of datasets did it work better with? What types of dataset did not not work well with?
5. **Extra Credit.** If you did any extra credit work, report it as well. You can insert extra credit reports as part of your narrative about specific datasets: e.g., if you implement multiple versions (see below), report multiple outputs and provide a comparison between them. Make sure that you specify which part is to be treated as extra credit. You can also put all your extra credit work into a separate **Extra Credit** section. In this case, any comparisons of your extra credit results with the regular results should be done in that section.
6. **Appendix. README.** Attach your `README` file with instructions on compiling and running your program as an appendix to your report.

Extra Credit

Experimentation with PageReank is subject to extra credit in the amount of 15% – 20%. To be eligible for the extra credit, your submission must have the following:

- Implement an extension of PageRank that tries to take in to account the score of the game / other edge label information.
- Make this extension customizable by a input parameter (collection of input parameters).
- Describe your extension in the README file.

To get the extra 5% of the credit you may also do the following with the **NCAA-FOOTBALL** dataaset:

- Try to find the optimal values of the custom input parameters that provide for the best, in your opinion, ranking of the (at least) top 25 - 40 football teams. Feel free to compare the rankings you obtained with the final regular season rankings by AP, Coaches, BCS standings, etc. Links to these rankings will be made available to you.
- Report your findings either in the README file, or in a separate micro-report file that you submit.

Deliverables and submission instructions

This lab has only electronic deliverables. Submit the following:

- Report.
- Source code for your program.
- `README` file (in addition to inserting it into the report).
- Any extra credit files.

Submit all electronic deliverables except for the report as a single zip of gzipped tar archive (`lab07.zip` or `lab07.tar.gz`). Submit the report as a separate file. The preferred format is PDF. Postscript, OpenOffice and MS Word 2000 (`.doc`) formats will also be accepted. MS Word Office 7 format (`.docx`) format **will not be accepted**. Use the following `handin` command:

```
$ handin dekhtyar lab07 <files>
```