Home Assignment 1.1

Version 1.1

Task 1

Crawl a two-mode network from a "real" social networking application of your choice (see Resources). "Real" means it should be an active web platform with active users such as *flickr*, *last.fm*, or *del.icio.us*. You can use RESTful or other APIs.

Briefly describe your choice and crawling strategy in your report `report.pdf`. Use *Python* to implement your crawler and submit your source code in `crawler/*.py`. You may use any library that is freely available on the internet, as long as you include the URL of where to get this library. Describe how to use your code in a text file `crawler/readme.txt`. The resulting network will be called network A from now on.

1. Make sure your network has the following properties:
   - The network is connected.
   - The average rate of participation is > 1.5.
   - The average size of events is > 1.5.
   - The minimum number of actors/events is 100/100. The maximum number of actors/events is 300/300.
2. Store the adjacency matrix of your network in a comma-separated file `networkA.csv`, where each row corresponds to an actor and each column corresponds to an event. Use commas (`","`) to separate values and single new-lines (`"\n"`) to separate rows. Example (3 actors, 2 events):

```
0,1
1,1
1,0
```

3. Store node labels in two UTF-8-encoded files `actorLabelsA.txt` and `eventLabelsA.txt`, such that each line contains one label, in the same order as in the adjacency matrix.
4. Import the network into Octave using `csvread`. This should result in an Octave variable `networkA`, an $m \times n$ matrix, where m is the number of rows/actors and n is the number of columns/events. (This is a preparatory step for Home Assignment 1.2.)

Task 2

Crawl an undirected one-mode network from a real social networking application of your choice (need not be the same one as in Task 1, but may be) in the same manner as in Task 1. This network will be called network B.

1. Make sure your network has the following properties:
   - The network is connected.
   - The minimum number of nodes is 100. The maximum number of nodes is 200.
   - The average degree is > 1.5.
2. Store your one-mode network as an adjacency matrix in a csv file `networkB.csv`, where rows and columns correspond to your nodes. Example (3 nodes):

```
0,1,0
1,1,1
0,1,1
```

(Note: Because B is undirected, this matrix has to be symmetric.)

3. Store node labels in a file `nodeLabelsB.txt`, such that each line contains one label, in the same order as in the adjacency matrix.

4. Import the network into Octave. This should result in an Octave variable `networkB` ($m \times m$).

Structure of your repository

- `report.pdf` (one page containing network choices and crawling strategy – keep it short!)
- `crawler/`
  - `main.py` (main program, shall be executable by invoking `python main.py`)
  - `*.py` (additional Python files)
  - `readme.txt` (UTF-8-encoded)
- `octave/`
  - `networkA.csv`
  - `networkB.csv`
  - `actorLabelsA.txt` (UTF-8-encoded)
  - `eventLabelsA.txt`
  - `nodeLabelsB.txt`

Your files `report.pdf`, `readme.txt` and every source code file has to have a header containing your name and matriculation number.

Submission

Home Assignment 1.1 is due **April 19, 2010 12:00** (high noon). You will build on your networks in Home Assignment 1.2.

The due date is a *soft deadline*. That is, your score on the assignment will be rated 100% if you hand in the assignment before 12:00. The following 12 hours are suitable for a submission as well, *but* your points will be rated 66%. Read: 1/3 of your points will be substracted if you hand in your assignment between 12:00 and 23:59. 24:00 is the *hard deadline*; if you hand in anything after 24:00 you will not receive any points.

Submission is done using the SVN version control system. (See instructions on the course website.)

Policies

- Your code and report will be checked for plagiarism.

Resources

- APIs listed on `http://www.programmableweb.com` and `http://code.google.com/intl/de-DE/apis/opensocial`.

- While you are free to select a social networking application of your choice, we recommend using one of the following APIs:
  - `http://www.23hq.com/doc/api/` (*flickr*-like, but does not require authentication)
  - `http://www.last.fm/api`
- Useful Python libraries:
  - `urllib2`
  - `xml.dom`
- Octave tutorials:
  - `http://www-mdp.eng.cam.ac.uk/web/CD/engapps/octave/octavetut.pdf`
  - `http://en.wikibooks.org/wiki/Octave_Programming_Tutorial`