SEWN 2015 - Assessed Coursework

Lab 4 - MSc version

Link analysis and computing PageRank

Ross Anthony - 13022691

Declaration

All code submitted as part of this assignment is my own work, with the exception of the following third-party libraries which are imported into the pagerank.py python script (see top of file). Any functions called elsewhere within this script which leverage these modules are not my work.

```
import math          # used for sqrt method

import numpy as np   # used for matrix data type

import json          # used to output data in json format for legibility

import urllib        # used to normalise URLs, with the unquote method
                       (i.e. convert %7e to ~)

from urlparse import urljoin # conversion of relative urls to absolute
```

Upon trying to run my PageRank algorithm with T = 0.00 it seemed to run indefinitely, well at least I didn't have the patience beyond waiting a couple of minutes for it to finish. Hence I chose to terminate the program and put in place a max number of iterations (of 100) to force it to reach an end. Examining the output in pagerank000_workings.txt (which contains the back trace of all PR calculations for each iteration) it appears the problem is setting the teleportation to 0.00 causes all of the PR values to eventually reach zero and therefore it never converges.

Out of curiosity I added some debug output which prints the URLs of any non-converged PageRanks to the screen as it's running. I also decreased the convergence threshold to +/-0.01. Previously it was set to 0.0001, which means the algorithm will keep on running and calculating new PageRanks if any of the PR values deviate more than 0.0001 from the previous iteration. Clearly this was too strict and should be loosened up, particularly when dealing with large matrices of links to avoid the algorithm running for an unnecessary number of iterations.

a. Do the results converge to the same value of PageRank no matter what the teleportation probability is?

No, the PageRank values vary quite considerably, although between 0.15 and 0.75 the overall weight of the PageRanks (i.e. their position in the leaderboard of PR's) is roughly the same. When running with either edge values for T, 0.00 or 1.00, the PageRanks seem meaningless.

b. What difference does the teleportation probability make to the number of iterations necessary?

For T = 1.00, all the PR values end up being 0.0022 after just 2 iterations. Whereas with T = 0.00 (with a convergence threshold of 0.01+/-), it runs for 56 iterations and the resulting PR values seem very unbalanced and incorrect given the in/outlinks.

c. Can you see any reason to choose 0.15 as a 'standard' value for teleportation?

It does appear as though the results are more accurate, or at least the seem inline with what one would expect to see in terms of the ordering of the PR's when considering the in and out links. If the T is too high it seems like it doesn't run for enough iterations for sufficient accuracy to kick in as it were.

d. Is there anything else interesting you can see in the pattern of results?

As alluded to above, there seems to be a general correlation between the teleportation factor and the number of iterations required to reach convergence. The lower the T value the the more iterations required and inversely the higher the T value the fewer iterations required.

Another factor which seems to influence the number of iterations required before reaching convergence is the level of precision of the decimal PR values. When I first began testing the algorithm I had the decimals set to the default in Python for floats, which is 18 decimal places. I found that to get the iterations under control (i.e. < 100) I had to round the values to the nearest 4 decimal places and set a convergence threshold of 0.0001+/-, although as mentioned above the later caused issues when testing it with T = 0.00 and the convergence threshold have to be loosen up to 0.01+/-.