Assignment 4: Web Crawling

Optional Bonus

Dalhousie CSCI 4173 — Winter 2007

1 Dates

Assigned: 13 March
Version: 21 March (5a)

Due: 05 April

2 Description

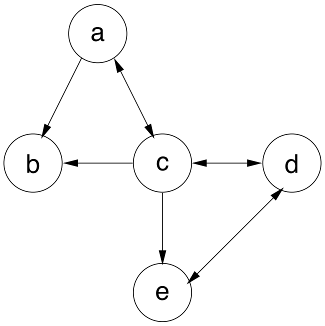
1. make a user-agent that will crawl the WWW following links to produce a site map of a particular website
   - the site map can be as simple as a list of webpages and what other pages they link to
   - the output does not need to be in XHTML (or HTML) form
   - a multi-stage approach (e.g. produce the map in text, import the text into a spreadsheet, and use the spreadshhet to compute the values) would be minimally acceptable
2. use the data from the sitemap to compute the compactness (C_p, see Equation 3), etc. of the website.

3 Help

To compute the in-degree, out-degree, compactness, etc. you must first create a *converted distance matrix (CDM)* from an adjacency matrix of the graph of webpages (nodes) and links (edges). The method was described in the lecture, and is in the articles cited there [1, 2][*].

[*] To read the articles from off-campus you will likely need to connect to the library proxy server (see ⟨URL:http://www.library.dal.ca/Find/Proxy/⟩ or ⟨URL:http://www.library.dal.ca/remote/proxy.htm⟩ for details).

The diagram and tables below (adapted from Rivlin et al. [2]) given an example of how to compute all of the basic values you will need.



	a	b	c	d	e
a	0	1	1	2	2
b	∞	0	∞	∞	∞
c	1	1	0	1	1
d	2	2	1	0	1
e	3	3	2	1	0

Shortest Path Matrix (M)

⇓

$$C_{r,c} = \begin{cases} M_{r,c} & \text{if } M_{r,c} \neq \infty \\ n & \text{otherwise} \end{cases}$$

	a	b	c	d	e	COD
a	0	1	1	2	2	6
b	n	0	n	n	n	20
c	1	1	0	1	1	4
d	2	2	1	0	1	6
e	3	3	2	1	0	9
CID	11	7	9	9	9	

Converted Distance Matrix (C)

$$COD_r = \sum_c C_{r,c} \tag{1}$$

$$CID_c = \sum_r C_{r,c} \tag{2}$$

$$C_p = \frac{(\max_C - \sum_r \sum_c C_{r,c})}{(\max_C - \min_C)} \tag{3}$$

3.1 Tips & Suggestions

- You may use public domain web crawler code for your assignment but you must acknowledge its source in compliance with the plagiarism policy. One candidate crawler you might want to adapt is the W3C's Link Checker (⟨URL:http://validator.w3.org/docs/checklink.html⟩).

- Any web crawler that you create or use should abide by the Robot Exclusion Standard so that you do not overwhelm any servers.

- Test your crawler on a small website in a restricted Internet domain (perhaps the examples subpart of the CS4173 site). Once you can print out the website hierarchy correctly then produce the array M, and from it, C. Once you have the converted distance matrix (C) you can compute the strata, and other properties of the website structure.

- Especially if you write your own crawler: plan for how to avoid cycles and showing multiple links from one webpage to another webpage — for every pair of webpages x and y, there should be at most one link from x to y in your sitemap.

4 Grading Scheme

Producing a basic sitemap alone will earn you a baseline grade (B flat) for the assignment. Anything you do beyond that will increase your overall points for this assignment.

5 Submission Instructions

Detailed instructions will follow. You will need to show your code and a captured run of your program.

References

[1] Rodrigo A. Botafogo, Ehud Rivlin, and Ben Schneiderman. Structural analysis of hypertexts: Identifying hierarchies and useful metrics. *ACM Transactions on Information Systems*, 10(2):142 – 180, April 1992. ⟨URL:http://doi.acm.org/10.1145/146802.146826⟩.

[2] Rhud Rivlin, Rodrigo Botafogo, and Ben Shneiderman. Navigating in hyperspace: Designing a structure-based toolbox. *Communications of the ACM*, 37(2):87 – 96, February 1994. ⟨URL:http://doi.acm.org/10.1145/175235.175242⟩.