Data Mining

Advanced Implementation Project

Dr. Jason T.L. Wang, Professor

Department of Computer Science

New Jersey Institute of Technology

Submission Rules

- Embed your last name and first name in your project file name. For example, if your name is John Smith, your file name should read: smith_john_advancedproj.doc. Only doc or pdf file is accepted.

  No tar/zip/rar is allowed.
- Your project will automatically lose **10** points if the above submission rules are violated.
- This is a single person project.
- Submit your project file in Moodle under Advanced Implementation Project Submission Site before the due time. The project file in Moodle is considered as the final version.
- No late project is accepted. A project is late if it is not submitted in Moodle before the due time. Zero points will be given to the late project.
- In addition, email an identical copy of your project file to TA with the same file name before the due time. If your project file is not emailed to TA before the due time, it is also considered late.

  Keep a copy of your email for verification when needed.

Advanced Implementation Project part 1

This web mining project is to implement the HITS algorithm for connectivity analysis. Use google to search for 30 web pages related to "Deep Learning.'' Create a root set that contains the 30 web pages, called seed pages. The root set is then expanded to a base set or neighborhood graph, by adding all of the pages that the root-set pages link to, and all of the pages that link to a page in the root set, up to a size cutoff threshold *k* where *k* is a user-specified parameter value. Each page in the neighborhood graph is initially assigned an authority weight of 1 and a hub weight of 1. Then use the HITS algorithm to iteratively update the authority weight and hub weight. Make sure to normalize the algorithm so that the weights converge. See

https://en.wikipedia.org/wiki/HITS_algorithm

for normalization details.

Advanced Implementation Project part 1

Your project should output the following.

(i) Print out the top N web pages with the largest authority weights. Rank these web pages in the descending order of the authority weights. For each output page, show its authority weight and hub weight.

(ii) Print out the top N web pages with the largest hub weights. Rank these web pages in the descending order of the hub weights. For each output page, show its hub weight and authority weight.

(iii) N is a user-specified parameter value. Your program should print out web pages for at least three different N values, namely N = 20, N = 10, and N = 5.

(iv) In addition, print out all pages in the neighborhood graph.

(v) Each page is printed/displayed in Google format, i.e., each page is displayed by its link/URL along with a few words describing the page.

Advanced Implementation Project part 2

Repeat part 1, but instead of building a neighborhood graph related to “Deep Learning,” you build a neighborhood graph related to “Machine Learning.”

Advanced Implementation Project part 3

Repeat part 1, but instead of building a neighborhood graph related to “Deep Learning,” you build a neighborhood graph related to “Data Mining.”

Advanced Implementation Project part 4

Repeat part 1, but instead of building a neighborhood graph related to “Deep Learning,” you build a neighborhood graph related to “Big Data Analytics.”

Implementation Notes

Each neighborhood graph must contain at least 100 web pages.

Your program must contain the source code for building the neighborhood graph and the source code for implementing the HITS algorithm. Building the neighborhood graph requires a web crawler that can traverse the web and gather pages from the web. The 30 seed pages can be collected manually or by your program.

Platforms are open

- Programming language is open; any one of the following is allowed: C, C++, C#, Java, R, Matlab, Perl, Python, Php, visual studio, PL/SQL, etc. Use any programming language of your choice (specify the programming language you use in the project).
- Operating system is open; any one of the following is allowed: Windows, Solaris Unix, Linux, Mac OS, etc.
- Hardware is open; any one of the following is allowed: PC, Laptop, Sun Sparc, etc.

Project Grading

- Your project file will be examined by me and TA. TA will email the grade of the project to you.
- Note: There is a limit on the file size in Moodle and in NJIT's email box. So, keep your project file small to avoid any problem that may occur when submitting the file in Moodle and emailing it to TA. The project file should contain the source code and documentation including **screenshots**. The screenshots are used to demonstrate the running situation of your program, particularly how the program executes and produces output based on different input data and user-specified parameter values.