11-741/11-641/11-441 Machine Learning for Text Mining
Homework 3: Link Analysis

TA in charge: Pengfei Wang

Due: February 18, 2016, 11:59 PM

In this assignment, you will be developing the PageRank algorithm and its personalized variants and compare them empirically on the provided CiteEval dataset.
This assignment consists of two major sections:

- Implementation: Write a program to compute PageRank, Personalized PageRank, and Query-sensitive PageRank for the documents in the CiteEval dataset. (Prefer Java/Python)
- Retrieval: Perform retrieval for the provided user-query pairs

Hint: It is important to do a good job on your report, so please start early to finish the program and leave enough time to write the report for each part. To simplify your work, a report template is provided.

1 Implementation

You will be implementing three PageRank algorithms.

- **Global PageRank (GPR)**
- **Query-based Topic Sensitive PageRank (QTSPR)**
- **Personalized Topic Sensitive PageRank (PTSPR)**. This is a personalized variant of QTSPR. Instead of weighing the topic-sensitive pagerank for each topic with the query-topic distribution, we weigh it with the user's interest in that particular topic. i.e. $\Pr(t|q)$ is replaced by $\Pr(t|u)$.

The transition matrix, the document classification information, user's topical interest distribution and query's topical distribution are pre-computed for you, refer to the "Dataset" section.
You may use any matrix multiplication library suitable for your programming language of choice or implement your own matrix multiplication routine.

Requirement: In your implementations use **0.8** as the dampening factor $(1 - \alpha)$.

Hint: you must strive to modularize your code to share the common components of the various PageRank algorithms. That will significantly reduce the programming effort.

2 Retrieval

For retrieval, you will be implementing variants of the Google's approach to combination of PageRank and search-relevance scores. One simple function is the weighted sum of the PageRank and search-relevance scores. Let's call this WS for weighted sum. In addition, you should also propose your own way of combining PageRank and search-relevance scores. Let's call this CM for custom method. Finally, you may refrain from using a combination of scores, and just use the PageRank scores to rank the documents. Let's call this NS for No-search as query-specific search results are not included.
The search-relevance scores of each query are pre-computed for you, refer to the "Dataset" section.
Based on the provided dataset, you will be comparing nine approaches to PageRank based search. The table below depicts these nine comparisons:

Method \Weighting Scheme	NS	WS	CM
GPR			
QTSPR			
PTSPR			

Thus, GPR will be implemented using three weighing schemes, NS, WS, and CM. Similarly for QTSPR and PTSPR, leading to nine total methods in your repertoire.
You will be reporting the following items for each of the compared approaches:

1. MAP (Mean Average Precision) averaged over all the queries
2. Precision at 11 standard recall levels (0%, 10%, ..., 100% recall levels) averaged over all the queries
3. Wall-clock running time in seconds (This should include PageRank computation time + retrieval time) averaged over all the queries.
4. Values of chosen dampening factor, weighing factor (for combining scores) or any other parameters in your system

You will be using the trec_eval program (online link here) to evaluate your system (very basic documentation here). trec_eval is widely-used, however it is extremely intolerant of format errors in its input. If you receive errors or don't get the results that you expect, the mostly likely reason is that the format is slightly wrong.
Additionally, you will also be analyzing the results, and stating your general observations about the various parameters in the system (4 above).

3 Evaluation Format

Your software must write results in a format that enables the trec_eval program to produce evaluation reports. trec_eval expects its input to be in the format described below.

`QueryID Q0 DocID Rank Score RunID`

For example:

```
10 Q0 clueweb09-enwp03-35-1378 1 16 run-1
10 Q0 clueweb09-enwp00-78-1360 2 11 run-1
10 Q0 clueweb09-enwp00-67-0958 3 9  run-1
:  :  :                        : :  :
11 Q0 clueweb09-enwp00-63-1141 1 18 run-1
```

The `QueryID` should correspond to the query ID of the query you are evaluating. `Q0` is a required constant. The `DocID` should be the external document ID. The scores should be in descending order, to indicate that your results are ranked. The `Run-ID` is an experiment identifier which can be set to anything.
When no documents are retrieved for a query, which may be the case for some structured queries, output a line with `'dummy'` as `DocID` and 0 as score for your trec_eval file. For instance for `query 10`, your dummy output should look like below.

`10 Q0 dummy 1 0 run-1`

4 Dataset

The transition matrix of the CiteEval documents is stored in *transition.txt*. This document is in the sparse matrix format. i.e. each row in the file corresponds to a non-zero cell of the matrix, e.g. a row of the form "`i j k`" denotes that the matrix contains a value `k` at the row `i` column `j`. The value `k`=1 denotes that there is a link from document `i` to document `j`.
The document classification information for TSPR algorithms is stored in *doc-topics.txt*. Each row is a `docid-class` pair.

User's topical interest distribution $\Pr(t|u)$ for all topics for PTSPR method is stored in *user-topic-distro.txt*. Each row is of the form:

$$a \ b \ 1 : p_1 \ 2 : p_2 \ldots\ldots \ 12 : p_{12}$$

where a is the `user id`, b represents the query by the user, and the items of the form t:pt denote topic probabilities $\Pr(t|a) = pt$
Query's topical distribution $\Pr(t|q)$ for all topics for QTSPR method is stored in *query-topic-distro.txt*. Each row is of the form:

$$a \ b \ 1 : p_1 \ 2 : p_2 \ldots\ldots \ 12 : p_{12}$$

where a is the `user id`, b represents the query by the user, and the items of the form t:pt denote topic probabilities $\Pr(t|b) = pt$.
The search-relevance scores of each query for retrieval part of the assignment is stored in *indri-lists.zip*. This zipped archive contains several files, (one per query), each containing the ranked-list returned by the Indri search engine for a particular query, with the corresponding retrieval scores. The document format is similar to the trec_eval format.

5 What to Turn In

5.1 Written report [60pts]

Submit your report in PDF format with the filename *HW3-YourAndrewId.pdf*. Please include your **name and Andrew ID** at the top of the first page of your report. Your report must contain the following sections, **each clearly labeled as an independent section**.

1. **Statement of Assurance:** You must certify that all of the material that you submit is original work that was done only by you. If your report does not have this statement, it will not be graded.
2. **Experiments**
   (a) Describe the custom weighing scheme that you have implemented. Explain your motivation for creating this weighting scheme.
   (b) Report the performance of the 9 approaches as described above.
   (c) Compare these 9 approaches based on the various metrics described above.
   (d) Analyze these various algorithms, parameters, and discuss your general observations about using PageRank algorithms
   (e) Discuss some general remarks about
   - What could be some novel ways for search engines to estimate whether a query can benefit from personalization?
   - What could be some novel ways of identifying the user's interests (e.g. the user's topical interest distribution $\Pr(t|u)$) in general?
3. **Details of the software implementation**
   (a) Describe your design decisions and high-level software architecture;
   (b) Describe major data structures and any other data structures you used for speeding up the computation of PageRank;
   (c) Describe any programming tools or libraries that you used;
   (d) Describe strengths and weaknesses of your design, and any problems that your system encountered

5.2 Sample Files [10pts]

Include the following files in your submission. Please use the given convention for naming them.

1. **GPR-10.txt**: GPR values after 10th iteration.
2. **QTSPR-U1Q1-10.txt**: QTSPR values of user 1 on query 1 after 10th iteration

3. **PTSPR-U2Q2-10.txt**: PTSPR values of user 2 on query 2 after 10th iteration

Each line in the file will contain a `documentID` with the calculated PageRank value. The format in file is

```
documentID PageRankValue
```

Note that there is a space in the middle not a tab.

5.3 Source Code [30pts + 10bonus pts]

A folder named **src** that contains the source code of the software you wrote for this assignment.
The TAs will look at your source code, so make sure that it is legible, has reasonable documentation, and can be understood by others. This is a Computer Science class lesson - the instructor will actually care about your source code. Please test your scripts and verify that your program can work on different datasets without any modification and that you include everything necessary to run it. Bonus points will be given for good implementation.
Please make it easy for the TAs to see how you have addressed each of the requirements described for each section.

6 Restrictions

1. Your system must do this task *entirely automatically*. You can use some parameters in command line to distinguish between different tasks. The TAs **will not** modify your source code in order to change the parameters or run a different experiment.
2. You must write all of the software yourself. No external PageRank related package is allowed.

7 Submission Checklist

Please compress all the following files into a .zip file for submission. Use **HW3-YourAndrewId.zip** as the filename.

1. All Source Code in **src** folder
2. Report HW3-YourAndrewId.pdf
3. Three Sample Files (GPR-10.txt, QTSPR-U1Q1-10.txt, PTSPR-U2Q2-10.txt)