11-741/11-441 – Machine Learning with Graphs **Due: March 19, 11:59PM**

Instructor: Yiming Yang

Homework 3: Link Analysis

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.
- Retrieval: Devising ways of combining PageRanks scores with the provided search-relevance scores to perform retrieval for the provided user-query pairs

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. Details for each of the deliverables follow. To simplify your work, a report template is provided. (It's okay to use LaTeX, but you must follow the same format as the template.)

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 or implement your own matrix multiplication routine.

Requirement: In your implementations use **0.8** as the dampening factor for the **document transition matrix** ($1 - \alpha$ for GPR and α for QTSPR, PTSPR as in the lecture).

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

In this assignment, you will be given the search-relevance scores for different queries. The goal is to implement different variants of the Google's approach to combine the PageRank scores with search-relevance scores for retrieval. One simple function is the weighted sum of the PageRank and retrieved search-relevance scores. Let's call this WS for weighted sum. In addition, you should also propose your own best 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.

Refer to the "Dataset" section for the format of pre-computed search-relevance scores.

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 after convergence:

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 for PageRank computation time and retrieval time, averaged over all the queries. For example,
   - GPR-WS: x secs for PageRank, y secs for retrieval.

   Hint: For GPR, it's okay to run the PageRank once and report the running time instead of running the PageRank repeatly for each query and take the average.
4. Values of weighing factor (for combining scores) or any other parameters which perform well in your system. (Note that you must fix the dampening factor for the document transition matrix as 0.8.)

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).

Hint: Due to the peculiarity of this particular dataset, you may find that in most cases, the PageRank scores are not helpful. You are not required to spend too much time trying to improve the performance. The aim is to analyze the effect of the weighting scheme and report your observation.

Hint: For each query, you only need to include the documents that are present in the provided search-relevance list, documents that do not appear in the search-relevance list can be ignored.

Hint: You should choose a reasonable and consistent error threshold as the stopping criterion. For example, stop the power iteration when $\|r^{(k)} - r^{(k-1)}\|_1 \leq 10^{-8}$.

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:

```
10-1 Q0 clueweb09-enwp03-35-1378 1 16 run-1
10-1 Q0 clueweb09-enwp00-78-1360 2 11 run-1
10-1 Q0 clueweb09-enwp00-67-0958 3 9  run-1
:   :  :                         : :  :
11-1 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.
You need to produce the ranking lists for the 38 search-relevance score files and combine them in one file with the correct QueryID for each line for evaluation. There should be totally 17,885 lines.

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-topicid` pair.
User's topical interest distribution $\Pr(t|u_{a,b})$ for all topics for PTSPR method is stored in *user-topic-distro.txt*, where $u_{a,b}$ denotes the system estimated profile for user a at the time the b-th query was issued. 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 b-th query by the user, and the items of the form t:p_t denote topic probabilities $\Pr(t|u_{a,b}) = p_t$.
Query's topical distribution $\Pr(t|q_{a,b})$ for all topics for QTSPR method is stored in *query-topic-distro.txt*, where $q_{a,b}$ denotes the b-th query issued by user a. 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 b-th query by the user, and the items of the form t:p_t denote topic probabilities $\Pr(t|q_{a,b}) = p_t$.
The search-relevance scores of each query is stored in the *indri-lists* directory. This directory 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 except the `QueryID` field is replaced by `0`. All the data files are provided in the data folder.

Hint: Files in *indri-lists* are named after convention *QueryID.results.txt*. You should replace the contents' `0` back to `QueryID` in your submission to trec_eval. QueryID is a-b where a is the `user id`, b represents the b-th query by the user. The user id, index of queries by user and topic id are 1 indexed i.e they start from 1 and assume the maximum document id to be the total number of documents present in the dataset. Note that b is used just for distinguishing queries from the same user. It does not imply that queries with the same b from different users are the same.

5 What to Turn In

5.1 Written report [60 pts]

Write your report in PDF format. 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, programming environment including the version and libraries that you used;

(d) Describe strengths and weaknesses of your design, and any problems that your system encountered

5.2 Sample Files [10 pts]

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

1. **GPR.txt**: Final converged GPR values.
2. **QTSPR-U2Q1.txt**: Final converged QTSPR values of user 2 on query 1
3. **PTSPR-U2Q1.txt**: Final converged PTSPR values of user 2 on query 1

Each line in the file will contain a `documentID` with the calculated PageRank value. You need to provide the scores for all documents, i.e, there should be 81,433 lines for each of the three files. The format of these files will be like

```
documentID PageRankValue
```

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

5.3 Source Code [30pts]

A folder named **src** that contains the source code and script to run 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 write your own script (hw3.sh) and verify that your program can work on different datasets without any modification and that you include everything necessary to run it. **Please make it easy for the TAs to see how you have addressed the requirements described for each section.** By default, the code will be tested on Ubuntu Linux 18.04 and Python 3.6.7, please make sure your code can run successfully under such environment. If you plan to use a different version please document the choice in a README.txt file. If you plan to use other programming languages, please discuss with TAs in advance.

6 Restrictions

1. Your system must do this task entirely automatically. No manual intervention is allowed. The TAs **will not** modify your source code in order to change the parameters or run a different experiment.

2. TAs will change the current directory into src and run your hw3.sh without any arguments and it should create all the three sample files, nine result files in the same folder and print the time taken for each of the 9 possible approaches on console. (i.e., do not overwrite the sample files in the parent directory.) The nine results files should be named as GPR-WS.txt, QTSPR-NS.txt, etc.
3. You must write all of the software yourself. No external PageRank related package is allowed.
4. You are encouraged to use Python 3 for the homework so that it's easier to verify your work under the same environment. It's okay to use scientific packages such as Numpy and Scipy for matrix multiplication, but you must write the PageRank algorithm by yourself. If you are unsure whether a particular API could be used, ask about it on Piazza. You must include the package dependency files such as requirements.txt or environment.yml if you used an external package.

7 Submission Checklist

The report and the source code should be submitted separately.

For the source code, please compress all the following files into a .zip file for submission.

1. All source code, script and data in **src** folder
2. Three sample files (GPR.txt, QTSPR-U2Q1.txt, PTSPR-U2Q1.txt)
3. (Optional) README.txt

The directory structure must follow the following format:

```
HW3.zip
├── src
│   ├── source files
│   ├── hw3.sh
│   ├── README.txt
│   └── data
│       └── [empty]
├── GPR.txt
├── QTSPR-U2Q1.txt
└── PTSPR-U2Q1.txt
```

Do **not** actually upload the **data** directory, but your program should assume the dataset exist under that directory when hw3.sh is executed.