CS 5683: Big Data Analytics

Assignment-6: HITS using Spark

Total Points: 20 (3% toward final)

Due date: Nov 15, 2020 at 11:59pm

In this assignment, we will implement the HITS algorithm in Apache Spark. You can use *numpy* operations, if necessary. You will experiment with a large real-world graph to find its node's hub and authority scores. **NOTE:** *This is an individual assignment.*

Dataset: Use the simple directed graph[1] given with this assignment to implement the HITS algorithm. The network is represented as edgelist, i.e. (*u v*) in the input file represent an edge from node *u* to node *v*. The given network contains 1,005 nodes and 25,571 edges.

HITS Implementation:

Assume the directed graph *G = (V, E)*, where *V* is a set of nodes and *E* is a set of edges. *G* has ***n*** nodes and ***m*** edges, all nodes have non-negative out-degree, and $L = [L_{ij}]_{n \times n}$ is an $n \times n$ matrix called *link matrix* such that for any $i, j \in [1, n]$:

$$L_{ij} = \begin{cases} 1 & if\ (i \to j) \in E \\ 0 & otherwise \end{cases}$$

Given the link matrix *L* and some scaling factors λ, μ, the hub score vector ***h*** and the authority score vector ***a*** can be expressed using the following equations:

$$h = \lambda L a$$

$$a = \mu L^T h$$

where ***1*** in above equations is the $n \times 1$ vector with all entries equal to 1.

Based on these equations, the iterative method to compute ***h*** and ***a*** is as follows:

1. Initialize ***h*** with a column vector (of size $n \times 1$) with all 1's
2. Compute $a = L^T h$
3. Compute $h = La$
4. Go to step 2

Note: Elements of ***a*** and ***h*** in the above algorithm should be scaled between 0 and 1

[1] http://snap.stanford.edu/data/email-Eu-core.html

Implementation in Spark:

Implement the above algorithm in Apache Spark. Set the number of iterations to *40*, assume that $\lambda, \mu = 1$, and then obtain the vectors $\boldsymbol{h}$ and $\boldsymbol{a}$. The link matrixes L and L^T can be very large and should be processed with only RDD. Whereas, the vectors $\boldsymbol{h}$ and $\boldsymbol{a}$ can be stored in disk or memory. Importantly, the data (edgelist) can be read only twice – once to find L and once to find L^T. The total number of nodes should be calculated while reading the data one of the two times. *However, this can be done with reading the data only once*.

NOTE: The given data should be considered as a directed graph.

Compute:

i. List of 5 top node ids and their hub scores
ii. List of 5 bottom node ids and their hub scores
iii. List of 5 top node ids and their authority scores
iv. List of 5 bottom node ids and their authority scores

Submission requirements:

1. Submit the python notebook (.ipynb) with your program and output
2. The python notebook should include your name

Grading Rubric: (Total – 20 points)

1. Data processing to find L and L^T – 5 points
2. Spark program to find $\boldsymbol{a}$ and $\boldsymbol{h}$ vectors: 10 points
3. Output: 5 points

Acknowledgement: This assignment is motivated from the Stanford's Mining Massive Datasets course assignment