TEL AVIV UNIVERSITY
Department of Computer Science
0368.3239 – Leveraging Big Data
Fall Semester, 2013/2014

Homework 3, Jan 8, 2014

- **Due on January 27, 2014.**
- **You are allowed to consult any sources or other humans, but you must write and submit the solution yourself and understand it in detail.**
- **Please submit a PDF file to the email address that appears on the Web site. The file should be named `firstname_lastname_HW3.pdf` . Keep a copy. Include your full name and ID in the file.**
- **Note that there are 150 point in total and the grade is out of a 100. Work that exceeds 100 will earn extra credit points.**

Streaming SVD: Liberty's Algorithm

1. **[40 points]** Let A be an $n \times d$ (n rows and d columns) matrix and let B be an $\ell \times d$ matrix, $\ell < d$.
a) Prove that

$$||A^tA - B^tB||_2^2 = \max_{||x||=1} (x^t(A^tA - B^tB)x)^2 = (\lambda_1)^2$$

where λ_1 is the largest eigenvalue of $A^tA - B^tB$ in absolute value. Recall that $||D||_2 = \max_{||x||=1} ||Dx||$.
b) Prove that among all $\ell \times d$ matrices B the one that minimizes $||A^tA - B^tB||_2$ is $B_{opt} = \Sigma' V'^t$ where $U\Sigma V^t = A$ is the SVD of A and Σ' is obtained from Σ by deleting all but the first ℓ rows and columns and V'^t consists of the first ℓ rows of V^t.
c) Consider the following version of the *frequent directions* algorithms of Liberty: (B^{i-1} is the approximation after i rows have been obtained.)
Initialize B_0 as an all zero $\ell \times d$ matrix.
Given the ith row a_i of A do

- **Set $B_+ \leftarrow B^{i-1}$ with the ℓth row replaced by a_i.**
- **Let $B_+ = U\Sigma V^t$ be the SVD of B_+.**
- **Let $\delta_i \leftarrow s_\ell^2$ where s_ℓ is the ℓth singular value in Σ.**
- **Let $\Sigma' = diag(\sqrt{s_1^2 - \delta_i}, \sqrt{s_2^2 - \delta_i}, \ldots, \sqrt{s_{\ell-1}^2 - \delta_i}, 0)$.**
- **Set $B^i = \Sigma' V^t$.**

Return B_n.

Let $\hat{B} = B_n$. What is the upper bound Liberty proves on $||A^tA - \hat{B}^t\hat{B}||_2$, please outline and explain the main steps in the proof (just the claims needed and how they relate to each other).

Triangles

Consider the graph that models the Facebook "Friends" relation: nodes correspond to users and there is an (undirected) edge (x, y) if and only if Friends($\{x, y\}$), that is, x and y are friends.

We are interesting in counting the number T of closed triangles the graph contains. A closed triangle is an (unordered) triple of nodes $\{x, y, z\}$ which satisfies Friends($\{x, y\}$) and Friends($\{x, z\}$) and Friends($\{y, z\}$).

Consider the following (sequential) algorithm for computing T. The algorithm uses some total order $\prec$ over nodeIDs. Say nodes have IDs $1, \ldots, n$ and we have $i \prec j \iff i < j$.

a. For each node i, generate all open triangles (paths of length 2) (j, i, k) in which i is the middle node, $j \succ i$ and $k \succ i$, $j \prec k$. (This can be done by examining each i and taking all pairs of neighbors such that $j \succ i$.)

b. For each open triangle (j, i, k) generated, check if the edge (j, k) exists. If so, count it as a closed triangle.

2. **[10 points]** Prove that the above algorithm is correct (counts each closed triangle exactly once.)

3. **[10 points]** Express a Map-Reduce form of the algorithm: Specify the key value pairs generated in each Map-Reduce iteration and the operations performed by the reducers.

4. **[10 points]** Two important parameters that affect the performance of the triangle counting algorithm are:

- $P2$: The number of open triangles that are generated.
- M: The maximum number of open triangles generated by one node (as a middle node).

a. Explain in what ways the values of $P2$ and M affect the performance of a sequential and of a Map-Reduce implementations (which resources are affected and how).

b. Give a simple example (a family of graphs with n nodes and total orders on the node of each graph) where the choice of order makes a significant difference in the values of $P2$ and M.

Try to make the gap as large as you can for a graph with n nodes.

5. **[10 points]** Download the file `facebook_combined.txt.gz` from the depository: `http://snap.stanford.edu/data/egonets-Facebook.html`.

The file is a subgraph of the Facebook "Friend" relation on approximately 4000 nodes. The file contains a list of undirected edges (each occurring once). The Web page contains further information on the file.

Plot the (reverse) cummulative degree distribution of the graph: For each node compute the degree (number of friends). Then plot for each integer i the ith largest degree, in the most informative way that you can. (**hint:** sometimes it is helpful to put both or one axis in a logarithmic scale.)

Describe in words your general observations on the degree distribution.

6. **[20 points]** Consider an application of the triangle-counting algorithm to the Facebook subgraph.

Compute $P2$ and M when using each of the following three total orders (hint: you can compute $P2$ and M without running the algorithm).

(a) nodeID (provided ID)

(b) lexicographic order: (degree, nodeID)

(c) lexicographic order: (5000-degree, nodeID)

Which order would you prefer to use ? explain.

7. **[10 points]** Using an algorithm of your choice, compute the number of closed triangles in the Facebook subgraph. Describe what you have done.

8. **[10 points]** For a node i, let $N(i)$ be the set of neighbors of i. The number of closed triangles can be expressed as

$$T = \frac{1}{3} \sum_{(i,j)\in E, i<j} |N(i) \cap N(j)| \ .$$

Consider the following general scheme for estimating T using Min-Hash sketches:

- Compute a Min-Hash sketch $S(i)$ of $N(i)$ for each node i.
- For each edge (i, j), use $S(i)$ and $S(j)$ to estimate $|N(i) \cap N(j)|$.
- Take $\hat{T}$ to be the sum over edges of the estimates of $|N(i) \cap N(j)|$, divided by 3.

Apply this scheme with bottom-k Min-Hash sketches: Given $h \sim U[0, 1]$, the sketch $S(i)$ includes the k smallest values) in $\{h(j) \mid j \in N(i)\}$. More precisely, $S(i)$ has the form $s_{i1} < s_{i2} < \ldots < s_{ik_i}$, where $k_i = \min\{k, |N(i)|\}$ and s_{ij} is the jth largest hash value of a node in $N(i)$.

Write an estimator for $|N(i) \cap N(j)|$ in pseudocode. Use the following guidelines:

- If both $|N(i)|, |N(j)| \leq k$ (We assume here that we computed the degree $|N(i)|$ for all i), we can compute the intersection exactly from the sketches $S(i)$ and $S(j)$.
- Otherwise, let $\tau = \min\{s_{ik}, s_{jk}\}$ (if $|N(i)| < k$ we define $s_{ik} \equiv 1$, if $|N(j)| < k$ we define $s_{jk} \equiv 1$). Let k' be the number of values in $S(i) \cup S(j)$ that are $\leq \tau$.

  Estimate the Jaccard similarity of $N(i)$ and $N(j)$ using the estimator in Lecture 3. Estimate the union $|N(i) \cup N(j)|$ using $\frac{k'-1}{\tau}$ (this is an inverse probability estimator). Then estimate the intersection size by the product of the Jaccard estimate and the union size estimate.

9. **[20 points]**

Use the file `http://www.cohenwang.com/edith/bigdataclass2013/Homework/facebook_rand.txt` to obtain a mapping of nodeIDs to "random hash" values in $[0, 1]$. The file contains in separate lines nodeID (integer between 0 and 4038) and "hash" pairs.

Compute bottom-k sketches of $N(i)$ for all nodes i using these "hash" values for $k = 10$.

Apply your estimator to estimate the number of triangles T. Compare to the exact value and compute the relative error. Discuss your conclusions/thoughts on the tradeoffs between work and accuracy of the approximate and exact solutions.

10. **[10 points]** Can you use the sketches to estimate the number of triangles incident to each particular node ? how ?