Data Mining Homework 3

Mining Data Streams

Group 22 - Blanca Bastardés Climent, Alice Ciallella

Short description

The implementation of this assignment has been done in Scala using Spark. All the code is provided in the notebook **HW3.ipynb**. The paper chosen for this assignment is Triest, which uses Reservoir Sampling as a sampling technique. We implemented the following classes:

- **BaseAlgorithm:** Triest base algorithm for triangles estimation using reservoir sampling.
- **ImprAlgorithm:** an improved algorithm for triangles estimation.

Instructions to run the code

We provide the following folders and files:

- **HW3.ipynb:** A Spark notebook containing code for triangles estimation in networks.
- **data/facebook_combined.txt:** file used for the task, which contains the edges of the networks representing friendships. The network has 4039 nodes (users) and 88234 edges (friendships).

Results

The size of the sample *S*, represented in the code by the parameter *M*, has value 5000. The algorithms give as output the estimation of the global and local number of triangles inside the network. Following the global estimation obtained from both algorithm, being the real number of triangles 1612010:

- **Base**: 1781352
- **Impr**: 1633637

Questions

1. **What were the challenges you have faced when implementing the algorithm?**
2. **Can the algorithm be easily parallelized? If yes, how? If not, why? Explain.**
3. **Does the algorithm work for unbounded graph streams? Explain.**
4. **Does the algorithm support edge deletions? If not, what modification would it need? Explain.**

1. We did not face major challenges while implementing the algorithms although choosing the optimal value of M took some time since we tried for different values to see how it influences the estimation. We choose the value of 5000 because it gives good estimation and short execution time (compared to 1/10 of the stream).

2. The algorithms cannot be easily parallelized because they rely on global counters to estimate the number of triangles globally and locally and on the sample, *S*. The use of multiple units implies the sharing of those variables, which is not feasible.
3. The Triest algorithms (Base - Impr) work for unbounded graphs because it gives the estimation of the triangles at any given time *t.* The fixed-size sample is created using reservoir sampling, which picks the elements with equal probability.

   However, when t >> M, the probability of picking a new element and discarding an old one is near zero. This means that after a while, the algorithm will base its estimation on the same sample without taking into account new ones. This might compromise the reliability of the algorithms for unbounded graphs.
4. The Base/Impr algorithms do not support edge deletions because they are insertion-only. To handle deletions, the algorithms should accept a dynamic stream as input, whose elements are composed of the edge and the operation (insertion, deletion), and two counters that store the unbalanced deletions and insertions. These variables should be considered before applying reservoir sampling to the new input and, in case of unbalances, the probability of the sampling is determined by the counters. The described modifications are already incorporated in an existing algorithm called *TRIEST Fully-dynamic*.