"Summary Assignment -1" (2022594)

Part A: Goal of this step is to preprocess the text file into a structured edge list, that can be used with graphframes...

data is first load to spark as raw text, it is filtered and cleaned. to remove whitespaces, and to retain only to and from info.

each valid line is then split to two tokens, first src, second destination, malformed rows are dropped. (empty or missing data)

Part B: This code first takes in all distinct ID's that appear in src, dst columns in part A.

It defines a graphframe g with the info. vertices ⇒ a column with distinct id's and edges src to dst info defining links..-

u maintains object info. for graph..-

the edges and vertices are counted using fnc's. available in graphframes...

groundtruth is same as values calculated / found

Parts C and D: the code finds wcc's and scc's in the graph formed in Part B...

It starts by creating a checkpoint file to ensure if failures occur they occur for jobs instead of entire task...

It then assigns a vertex ID label for its wcc, treating edges undirected.

then it counts how many vertices fall into each component and sorts by size then picks largest wcc. by querying.

finally it counts edges in wcc for this it joins with a known path similar to a recursive job, then filters to edges that were given largest src id's and dest id's to give edges in largest wcc.

Same approach for scc.

Part E: We first build on a simple undirected graph by removing loops and dropping duplicates.

degrees: number of neighbors per node

triangleCount: for each node how many triangles it belongs to.

We join all vertices as per degree and triangle count ... ~

∃ node if degree $\leq 2$ : local clustering defined as 0.

else: $$\frac{2t_i}{d_i(d_i-1)} = C_i$$

$d_i \rightarrow$ degree of $i^{th}$ node

$t_i \Rightarrow$ triangle count of $i^{th}$ node.

Avg clustering coeff:

$$\bar{C} = \frac{1}{|V|} \sum_{i \in V} C_i$$

total $\Delta$ count:

$$T = \frac{1}{3} \sum_{i \in V} t_i$$

total wedges:

$$\tau = \sum_{i \in V} w_i = \sum_{i \in V} \binom{d_i}{2}$$

$$= \sum_{i \in V} \frac{d_i(d_i-1)}{2}$$

finally o/p: $T/\tau$ $\Rightarrow$ answer.

"Part F": in trying to compute the graph diameter and effective diameter (90th %ile shortest path length)

∵ graphframes is directed but we define it for undirected we add edges to make the graph undirected.

We first compute giant (largest) CC. then we take edges only in LCC., At this point we still store edges one way.

Now we duplicate each edge ie. add (u,v) for ∀ (v,u)

Now build g_lcc_sp = Graphframe on

v_lcc + g_edges_bi

then we choose landmarks L=256 to approx. dist efficiently

we then compute shortest distances to landmarks it can reach... we explode map into rows and drop dist == 0 pairs.

↓

self pairs.

finally diameter is max of all distances.

eff. diameter = 0.9 x total

find 1st bin where CDF ≥ target and linearly interpolate b/w that and prev. bin.

$d_{0.9} = \inf\{ d \mid CDF(d) \geq 0.9 \}$

explanation of results and diff. in results:

| | Metric | GroundTruth | Computed | AbsDiff | RelDiff_% |
|---|---|---|---|---|---|
| 0 | Nodes | 7115.00000 | 7115.000000 | 0.000000 | 0.000000 |
| 1 | Edges | 103689.00000 | 103689.000000 | 0.000000 | 0.000000 |
| 2 | Largest WCC (nodes) | 7066.00000 | 7066.000000 | 0.000000 | 0.000000 |
| 3 | Largest WCC (edges) | 103663.00000 | 103663.000000 | 0.000000 | 0.000000 |
| 4 | Largest SCC (nodes) | 1300.00000 | 1300.000000 | 0.000000 | 0.000000 |
| 5 | Largest SCC (edges) | 39456.00000 | 39456.000000 | 0.000000 | 0.000000 |
| 6 | Avg. clustering coefficient | 0.14090 | 0.140898 | -0.000002 | -0.001529 |
| 7 | Number of triangles | 608389.00000 | 608389.000000 | 0.000000 | 0.000000 |
| 8 | Fraction of closed triangles | 0.04564 | 0.041826 | -0.003814 | -8.355865 |
| 9 | Diameter | 7.00000 | 7.000000 | 0.000000 | 0.000000 |
| 10 | Effective diameter (90-percentile) | 3.80000 | 3.765724 | -0.034276 | -0.901990 |

in 6,

Reason: spark's Δcount + degrees are aggregated in 11 & 1 flop Σ order differs slightly from ref. implementation, causing rounding drift, numerical noise.

in 8, T matches so error in Σ code sums wedges in all nodes Ground truth typically only largest component.

including degree 1 and 0 inflates den. a little

In (0, using L = 256, approx. all shortest paths, sampling var can slightly over or undershoot %iles.

eff. diameter is sensitive around CDF threshold (90%), if one bin gets extra counts, interpolation might shift.

increasing L can reduce var. more.