CSE 5800 Mining/Learning and the Internet—HW4
Due Nov 12, Wed, 6:30pm
Submit Server: course= ml-internet , project=hw4

1. Implement BridgeCut with four versions:
   - (a) edge with the highest Bridging Centrality ($C_{Br}(e)$ in the paper)
   - (b) vertex with the highest Bridging Centrality ($C_{Br}(v)$)
   - (c) edge with the highest Betweenness ($\Phi(e)$)
   - (d) vertex with the highest Betweenness ($\Phi(v)$)
2. Allow this parameter:
   - (a) density threshold (densityThreshold in the paper)
3. Measure performance using:
   - (a) Davies-Bouldin index
   - (b) Silhouette Coefficient
4. Use three groups of data sets:
   - (a) toy data sets on the course web site
   - (b) real data sets on the course web site
   - (c) your own data set
5. Disscuss in a report (in pdf):
   - (a) Sensitivity analysis of parameters using enron2.txt:
     - i. vary density threshold
     - ii. calculate each performance measurement,
     - iii. plot performance vs. density threshold
     - iv. discuss the value for density threshold that seems to achieve the highest performance.
   - (b) Compare the algorithms using enron2.txt:
     - i. plot performance vs. density threshold for different algorithms
     - ii. plot performance vs. number of clusters for different algorithms (different density thresholds generate different number of clusters)
     - iii. plot clustering coefficient vs. number of nodes (edges for the edge-based algorithms) deleted (up to top 20) for different algorithms [Figure 5b in the paper]
     - iv. plot number of singletons vs. number of nodes (edges for the edge-based algorithms) deleted (up to top 20) for different algorithms [Figure 5d in the paper]
     - v. discuss the relative performance of different algorithms
6. Implementation:
   - (a) preferably use one of these programming languages: C, C++, Java, Python, or LISP.
   - (b) input file: a file for vertices and edges
   - (c) two modules:
     - i. BridgeCut: input graph; output:
       - top edge/vertex when it is removed
       - for each cluster, output vertices in the cluster
     - ii. Evaluate: input vertices and cluster membership; output performance
7. Submission:
   - (a) source code
   - (b) your data set
   - (c) report in pdf
   - (d) README.txt (how to compile and run your program/experiments on code.fit.edu or hopper.cs.fit.edu)