COMP26120 Lab 10

Tian Breznik

March 6, 2019

1 The small-world hypothesis

The small-world hypothesis states that in any large graph (network) any pair of nodes is connected by a short path. In our case, networks being networks of friends and nodes representing people. This statement is very counter-intuitive and works against our understanding of distances in networks. I hypothesize the statement is wrong as, the scale of the world outweighs the impact of individuals. Historically, looking at evolution, we can make a logical statement that communities did not mix due to environmental, cultural and other reasons. So the small-world hypothesis seems to lean towards humans having a common ancestor or globalisation, which has become significant in building communities only in recent centuries and its effect still fades in rural areas. The notion of attributing the small-world hypothesis in networks of connected people is also absurd as ancestry trees grow exponentially. We can test this statement by applying a shortest path algorithm between all pairs of nodes in the graph and analyse the results based on some chosen constant. We can record various statistical parameters like the mean "distance" between pairs of friends, as well as the variance of the data which together with the mean gives us an idea on the distribution of the data thereby providing evidence to the truth of Milgram's statement. In my opinion, the optimal distribution of the data which would give the greatest weight to Milgram's claims would be some mean near 6 degrees of separation with small variance values, which would mean that the data is dispersed locally near the mean and there is a small probability of having outliers far from the mean.

2 Complexity Arguments

The general complexity of Dijkstra's algorithm is O(Tm-h + nTdeq + mTd-k) where Tm-h, Tdeq and Td-k are the costs of make-heap, dequeing and decreasing a key. Since a priority queue is used in our implementation the complexity becomes O((m+n)logn), where m is the number of edges and n is the number of nodes. In our case m¿n so the complexity becomes O(mlogn).

The complexity of Floyd's algorithm on the other hand is O(n^3) and it is used for finding the shortest path between all pairs of nodes in a graph and works regardsless of the presence of negative-weighted edges, which break Dijkstra's algorithm.

However edges in a social network aren't weighted and we can find all shortest paths between nodes with Dijkstra's algorithm by running it on all nodes with time-complexity O((n*m)logn) which can become O(n^3) if the graph is dense with m = O(n^2). The social network graphs in our exercise are far from dense. So in our case Dijkstra beats Floyd's algorithm in terms of time complexity.

3 Part 2 results

Why Breadth-first search finds the shortest paths on an unweighted graph, and hence why is Dijkstra's algorithm not needed in this case. Breadth-First Search (BFS) just uses a queue to push and pop nodes to/from. This means that it visits nodes in the order of their depth. Whereas in Dijsktra's algorithm a priority queue has to be maintained with respect to the weights of the edges, this loses all significance with unweighted graphs because the priority queue cannot be maintained anymore with respect to the weights and hence a queue that is maintained by order of depth of a node is enough to ensure shortest paths will be found. In Dijkstra's algorithm a node is removed from the priority queue when a shortest path to it has been found, which requires the priority queue being sorted and our implementation having a removeMin() method. In breadth-first search a node is removed from the queue when it is found, this also guarantees that the shortest distance to the node has been found as BFS traverses the graph by breadth so the first occurrence of a node will be the one closest to the source node.

The results of the experiments are presented in the table bellow:

Data	Nodes	Edges	Time (s)	Mean	Variance
Caltech	769	33312	0.61	2.314	0.427722
Oklahoma	17426	1785056	2678.59	2.767	0.367257

4 Part 3 results

In part 3 of the exercise, I ran both breadth-first search and the heuristic algorithm on the same data and recorded the results of the distances in two matrices. Because the breadth-first search algorithm is guaranteed to find the shortest distances between the nodes in an unweighted graph, I compared its results to the results of the heuristic algorithm, which gave me the accuracy of the heuristic algorithm in finding shortest paths. The heuristic algorithm was not as successful as I initially thought in finding the shortest distances. Namely, on a graph with around 250 nodes it gave an accuracy of about 34.6seems very unreliable in terms of practical applications. Unfortunately, I haven't managed to test the heuristic with larger data like Oklahoma.gx, because the heuristic algorithm would run for about 5 hours on it. Although I hypothesize that the success of the heuristic algorithm would diminish with larger data sets, as the probability of a target node appearing on an arbitrary path grows larger and larger.

Another problem with the proposed heuristic algorithm is that, we mark the target node as unreachable if all the children of a found maximum node are already explored. The better alternative to this would be to find the node with the second largest outdegree after the node with the maximum outdegree has been explored, and explore that one.

5 Conclusions

The results in part1 show that the small world hypothesis might be close to reality in a University network. Both in Oklahoma and Caltech the mean and variance give strong evidence towards the small world hypothesis. The mean and variance show relatively low discrepancy in the data with relatively few outliers, which were at most 10 in both data sets. But these results cannot be generalized as Universities usually have their own Alumni networks which are well-connected to other universities, this creates a relatively close-knit Academic community. I would assume that the experimental results would not support the hypothesis to the same degree if we were to test it on data sets of random people from random countries or even from within the same country or city, people generally form isolated communities of friends and it is not likely for these communities to have a lot of common individuals. However the running-time data supports the proposed complexities.

The results from part3 show that if we are given a graph continually changing and have to resort to heuristic algorithms, we will have to sacrifice some accuracy, as the used algorithm is no longer mathematically rigorous. In our case it can't be proved that the heuristic algorithm will find shortest paths, however we can optimize the algorithm to increase its success rate in finding shortest paths.