Graph Algorithms Capsule Specification

Purpose

This capsule provides a group of graph algorithms that leverage the power of InfiniteGraph to perform common or expensive tasks. The algorithms can be positioned as samples or as an embryonic graph analytics framework.

Functionality

Overview

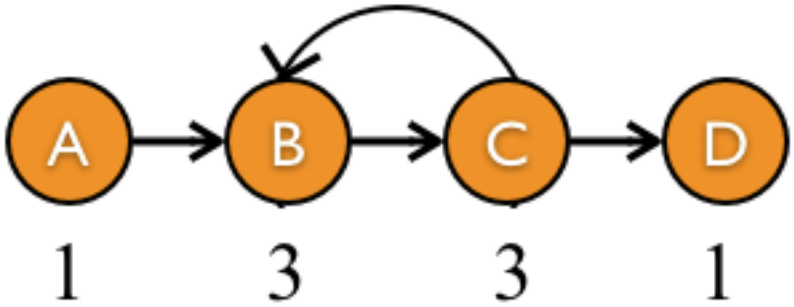
InfiniteGraph already supports conditional transitive closure navigation and shortest path finding. This capsule provides methods that support the following algorithms:

1. **Node Degrees:** Compute the node degree for a set of nodes.
2. **Degree Distribution:** Compute the distribution of Node Degrees for the whole graph. [This could leverage the parallel task execution feature of Objectivity/DB or MapReduce.]
3. **Connectedness:** Check whether a set of nodes are directly or indirectly connected in the graph. [This leverages the existing relationship analytics capabilities of InfiniteGraph and Objectivity/DB].
4. **Degree Centrality:** Compute the degree centrality for a set of nodes.
5. **Closeness Centrality:** Compute the closeness centrality for a set of nodes.
6. **Betweenness Centrality:** Compute the betweenness centrality for a set of nodes.
7. **Bridges:** Find bridges in the graph.
8. **Graph Diameter and Average Path Length:** Compute the diameter (span) and the average path length of the graph.
9. **Mutual Associates:** Determine whether a set of nodes can be connected via a node that is either a part of the set or is at most one link away from the members of the set. There may be multiple such nodes.

Each of the algorithms is now described in more detail.

1. Node Degrees

The degree of a node in a network is the number of connections it has to other nodes. In the diagram below: A and D each have a node degree of 1; B and C each have a node degree of 3.



The algorithm counts the total number of Edges connected to a particular Vertex.

2. Degree Distribution

The **degree distribution** is the probability distribution of the degree of each node over the whole graph. The algorithm should provide methods to compute: the Probability Mass Function (PMF) of each node degree value; plus the mode, median, mean; variance, standard deviation and skewness of the PMFs.

3. Connectedness

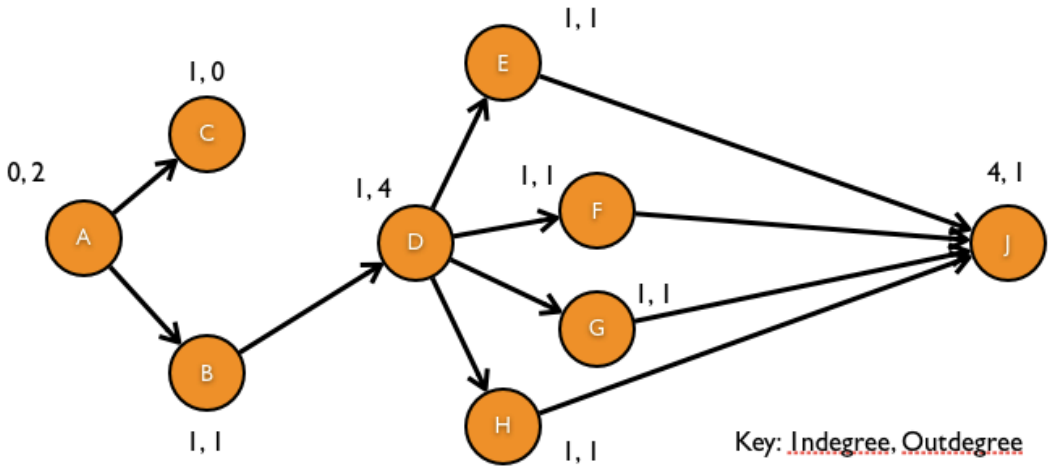
Determines whether all of the nodes in a given set are directly connected to one another without the involvement of other nodes or can be indirectly connected via an additional set of nodes. In the diagram above, besides the individual connections: A, B, C and D are connected; A and C can be connected via B; A and D can be connected via B and C; and B and D can be connected via C.

4. Degree Centrality

In general, centrality is an assessment of the importance of a node within a network. Degree centrality is the simplest, being a count of the number of connections that a node has. It would be the same as the Node Degree, but for the fact that it is generally separated into two figures, the number of incoming connections (indegree) and the number of outgoing connections (outdegree).

In a computing network, a node with a large indegree count could be attacked from many directions. Likewise, a node with many outgoing connections (outdegree) would be a good place to insert malware to broadcast it to other nodes.

The diagram below shows a directed graph (whose links can only be traversed one way) where J has a large indegree count and D has a large outdegree count.



The algorithm will allow the user to:

a) Specify the types of Edge to be regarded as “in” or “out” connections and to compute the indegree and outdegree values for a particular Vertex.

b) Locate the vertex or vertices with the highest indegree, highest outdegree or highest degree centrality in the whole graph.

c) Locate the vertex or vertices with the highest indegree, highest outdegree or highest degree centrality in vertices connected to a particular vertex or set of vertices

5. Closeness Centrality

Closeness considers the shortest paths between nodes and assigns a higher value to nodes that can be used to reach most nodes most quickly. As an example, consider a hub and spoke structure, where node A is connected to B, C, D, E, F etc. and those nodes are not connected to any others. Any node can be reached from A in one hop. It takes two hops to get from any other node to any node other than A, so A has the greatest centrality. Closeness is a good way to find the most powerful “influencers” in a social network.

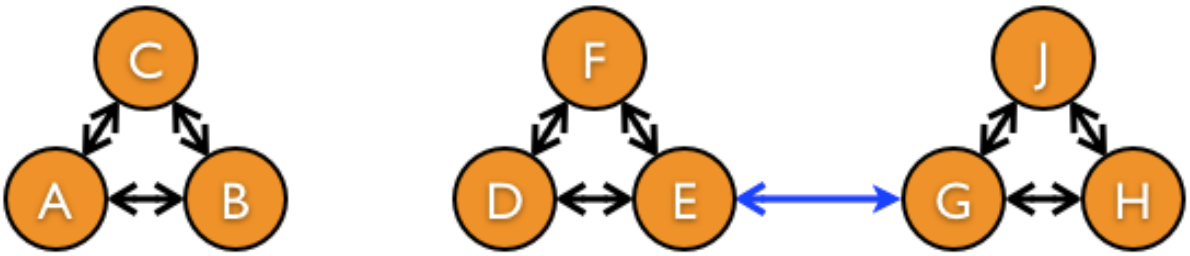
There are various ways to compute the closeness centrality. The algorithm will compute the smallest number of hops needed to reach all other nodes in the graph from a particular node. It will also find the node or nodes with a total hop count less than that of all other nodes.

6. Betweeness Centrality

Betweenness is a centrality measure of a node within a graph. Nodes that have a high probability of being visited on a randomly chosen shortest path between two randomly chosen nodes have a high betweenness. This concept of betweenness is often used to assess the influence of an individual on a social network. The algorithm will compute the betweenness of a particular node.

7. Bridges

A subgraph in which any two nodes in it are connected to each other by paths (of Edges and Vertices) and where none of the nodes on those paths are connected to "external" nodes is an island. If there is a connection to another node then the connection is known as a bridge. For example, A <-> B <-> C has no bridges. The graph below, consisting of subgraphs D <-> E <-> F <->D and G <-> H <-> J <->G, has a bridge between E and G. If the bridge were removed then the two subgraphs would be islands.



The algorithm will look for each Edge that connects two Vertices that are parts of subgraphs that would become islands if the Edge were removed. The algorithm will sweep the whole graph or be constrained to a designated subgraph. The subgraphs may be static or be conditional upon the application of query predicates.

8. Graph Diameter

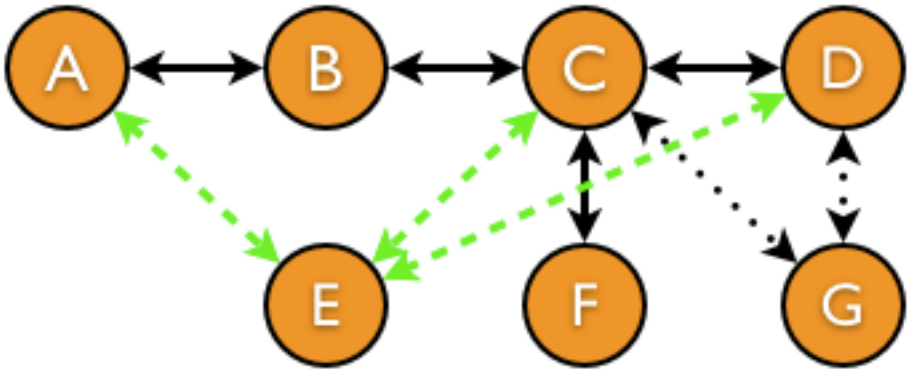
The graph diameter is the maximum distance between any pair of vertices in the graph. It can be computed by finding the lengths of the paths between each pair of nodes and selecting the highest value. The maximum path length is a measure of the diameter of the graph. The diameters of the two graphs immediately above are 2 and 5.

9. Average Path Length

As its name suggests, this is established by finding the shortest path between each pair of nodes and then computing the average of all of the path lengths.

10. Mutual Associates

The algorithm will find common nodes that are connected to all members of a group of nodes. In the graph below, node E is a Mutual Friend of nodes A, C and D. Likewise, node G is a Mutual Friend of nodes C and D. Node B could become a part of the social network consisting of A, C, D and E if B were to be connected to E. This algorithm could be used to supply "Friend" suggestions to people in a social network.



Platforms and Languages

- Java and C++ (later, if InfiniteGraph supports it) on Linux, Windows, Solaris and Mac OS X.