**netzen** - a software tool for the analysis and visualization of network data


---

**Exercise**

In this exercise, you will use the data set in the directory `/YourNetzenFolder/examples/vast/`. To load the data set, run the following command line under `/YourNetzenFolder/`:

```
./netzen examples/vast.script
```

Check the *Info* tab to see if the node table `vastnodes.txt` and the edge table `vast.txt` are both loaded. There should be 400 nodes and 19,670 edges that are loaded. Then go through this exercise and answer the following questions.

1. **Identify the five most important nodes by the degree property of nodes. Use a scatter plot for this problem.**

   **Answer**

   One way to find out nodes with high degree is to use scatter plots. Create a new scatterplot first:

   - Click View–Create–ScatterPlot. Then in its the property table:
     - set *Data.x* as the node property *NodeId*;
     - set *Data.y* as the node property *Degree*;
     - adjust *pointSize* to 6.
Nodes that scatter at the top of the plot are those with high degree, which have large number of edges linked to or linked from. In this way, we can treat a node with high degree as important. To show the ID of these important nodes:

- Set Label as the node property NodeID (deselect Default and then select NodeID in the pop-up multiple selection window);
- Move your cursor to those nodes at the top of the scatter plot to show their ID.

See Figure 0.1 for example.

From the scatterplot, the five most important nodes are 1, 309, 5, 306, and 0.

2. Can you identify other important nodes? This time, try to compute EigenCentrality and identify nodes with high centrality in your views.

Answer

Compute EigenCentralityEigen:
• Click the Analysis Tools tab – Centrality – EigenCentralityEigen
• Set “Edge Weight” as “Multiplicity” to consider the multiple edges that link to the same pair of nodes;
• Hit Go button. Computation is done when the progress bar hit 100%.

Set Y axis of the scatterplot as this newly computed property.

• Set “Data.y” to “EigenCentralityEigen” in the property table of Scatterplot under the Inspector tab.

Now you can hover nodes in the scatterplot to check the NodeID of the top most five nodes in the scatterplot NodeId-EigenCentralityEigen. They are 1, 5, 0, 13, and 309, as labelled in Figure 0.2.

Moreover, you can relate Degree to the node color in the same scatterplot, so that the two type of importance property can be visualized in the same plot. In the property table of the scatterplot,

• Set “color” to “Degree”
• Set “colormap” to “YlOrBr”.

So you can see that Node #13 has higher value in EigenCentralityEigen then Node #309, but it has smaller degree than the other shown by its lighter color.

3. Let’s find out high centrality nodes in a node-link diagram. *Hint: use GraphView.*

**Answer**

Create a view for a node-link diagram:

• Click View–Create–GraphView
• Set Node color as EigenCentralityEigen
• Set Node colormap as YlOrBr
• Set Nodesize as 0.01

Create a force-directed layout tool:

• Click Layout–Create–ForceDirected
• In the property table of GraphView, set “Layout” as “Force Directed_1”
• Now enable the layout. In the property table of ForceDirected_1, set “Enabled” as “true”.

Then the nodes in the graph move iteratively according to the parameter setting in the Force Directed_1. After a while, the graph looks like Figure 0.3. You can see the nodes with high degree are surrounded by dense edges, and the darker color (brown) highlights nodes with high Eigen Centrality. Important nodes are around the center of the graph.
Figure 0.3: Use GraphView to look at the dataset.

4. Try to simplify the network. *Hint:*

   – compute Betweenness centrality of edges in the graph (set “Edge Weight=Multiplicity”).
   – filter out edges with low centrality (set “EdgeThresholdProperty=Betweenness.Edge” and threshold 0.08).
   – re-apply the force-directed layout with the same edge threshold.

(a) Can you see any structure after simplifying the graph?

(b) This network consists of cell phone communication between leaders of a terrorist organization. Can you point out any groups or factions in the network?

*Answer*

Compute Betweenness:

- Click *Analysis Tools–Centrality–Betweenness*
• Set *Edge Weight* as *Multiplicity*.

• Hit *Go*

Filter out edges of low betweeness in the graph

• Click *Inspector–GraphView–GraphView_1*;

• Set *EdgeThresholdProperty* as *Betweenness.Edge*;

• Increase *EdgeThreshold* from 0.00 to 0.08.

This removes edges with little betweenness, i.e. the less ”central” communication paths. The remaining edges should be the ones through which most of the communication goes, i.e., represented the backbone of the network.

Still the graph is not quite clean. Let’s apply the same edge threshold setting to the layout:

• Click *Layouts–ForceDirected–ForceDirected_1*;

• Set *EdgeThresholdProperty* as *Betweenness.Edge*;

• Increase *EdgeThreshold* from 0.00 to 0.08;

• If the layout is disabled, set *Enabled* as *true* to start the layout.

The disconnected nodes drift away and the simplified network remains in the center of the view. Set *Label* as *NodeID* to show node label. Resize the node if necessary. After layout the structure of the network can be seen as shown in Figure 0.4.

From the important nodes you identified in the previous problems, you can see that:

• 306 - 309 - 0 form a communication triangle. Each of them are the hub of the group surrounded them;

• 0 - 1 - 5 the three nodes of high centrality (seen from the node color) form another faction.
Figure 0.4: Simplified network by Betweenness.
But these two communication groups are different in that [306 - 309 - 0] communicate directly, where as [0 - 1 -5] communicate indirectly via nodes 4 and 13.

5. Let’s simplify in a way that we get a global view, but preserve the structure as well. Hint:

– compute \( \text{EigenCentralityEigen} \) (set “Derivatives=true”) for edges;
– compute Minimal Spanning Tree (set “Weight=\( \text{EigenCentralityEigen}.\text{Derivative} \)” “Skeleton Property=Betweenness.\text{Edge}”, and “\( \text{Skeleton Threshold}=0.2 \)”);
– visualize Minimal Spanning Tree in GraphView by filtering out edges according “\( \text{EdgeThresholdProperty}=\text{inMST} \)” and “\( \text{EdgeThreshold}=1.0 \)”;
– to emphasize nodes with high degree, in GraphView set “Node opacity=Degree” and “\( \text{Node Transparency}=2.0 \)”;
– re-apply the layout by “\( \text{EdgeThresholdProperty}=\text{inMST} \)” and “\( \text{EdgeThreshold}=1.0 \)”.

(a) Can you see how the important nodes connect to each other?
(b1) What can you tell about the interaction between nodes 1, 5, 3 and 2?
(b2) What about the interaction between nodes 309, 306, 360 and 397
(c) If I were to tell you that these two groups (in b1 and b2) are in fact different cell phones, and that a single person has two devices, one of each group, can you point out what these pairings are?

\underline{Answer}

Compute EigenCentralityEigen again for edges:

• Click Analysis Tools–Centrality–EigenCentralityEigen
• Set Derivatives as true this time, so that centrality for edges are computed.
This computation will take longer than in problem 2.

- Set *Edge Weight* as *Multiplicity*.
- Hit *Go*

Compute Minimal Spanning Tree by the EigenCentralityEigen and filtering by Betweenness:

- Click *Analysis Tools–Graph–Minimum Spanning Tree*:
  - Set *Weight* as *EigenCentralityEigen Derivative*;
  - Set *Skeleton Property* as *Betweenness.Edge*;
  - Set *Skeleton Threshold* as 0.2.
  - Hit *Go*

This process generates a subgraph that spans from the minimum spanning tree of the graph, but adds important edges (given by the skeleton property). To visualize the Minimal Spanning Tree, filter out edges according to inMST property by the following steps:

- Click *Inspector–GraphView–GraphView.1*:
  - Set *EdgeThresholdProperty* as *inMST*;
  - Set *EdgeThreshold* as 1.0;
  - Set *Label* as *NodeId*;
  - Set *Node opacity* as *Degree*;
  - Set *Node Transparency* as 2.0;

And then re-apply the layout filtering out edges by inMST:

- Click *Layout–ForceDirected–Force Directed.1*:
  - Set *EdgeThresholdProperty* as *inMST*;
Figure 0.5: Simplified network by Minimal Spanning Tree.

- Set Enabled as true;

The resulting graph should show interesting structure, as seen in Figure 0.5.

Now the important nodes are more opaque (according to degree) so we can easily find out nodes with high degree. This make those leaf nodes with degree one barely visible and faded out in the graph. Note that high degree nodes can also be identified by the edges around it as well.

The group [306 - 309 - 0] can be seen in this graph too, where as the other group triangle [0 - 1- 5] is more apparent now compare to it in the previous problem.

(b1) What can you tell about the interaction between nodes 1, 5, 3 and 2? We can see communication between these pairs: 1–2, 1–3, and 1–5. Also, 1 and 5 are hubs that have fairly high degree.

(b2) What can you tell about the interaction between nodes 309, 306, 360 and 397?
Communication can be identify between these following pairs of nodes in the graph: 309–306, 309–397, and 309–360. Note that there is no link between 360–306 and 397–306.

(c) pairings of cell phones that belongs to a person

(1, 309) should be the same cellphone holder because they both are the most active and high degree nodes in the groups.

306 should belong to the same holder of 5, considering that they both call 0 and 21. Similarly, we may conclude that 397 should be held by the one who has 2 but not 3. Because 397 and 2 both call 54 and 309 (which is 1 too). This gives us the pair (397, 2).

Then the rest pair is (360, 5) that both call the person having (309, 1).

In sum, judging by the symmetry of the connectivity, these nodes are always connected via other nodes, and when considered along they preserve a hierarchical structure.
Figure 0.6: An example of integrated views in netzen for this exercise.