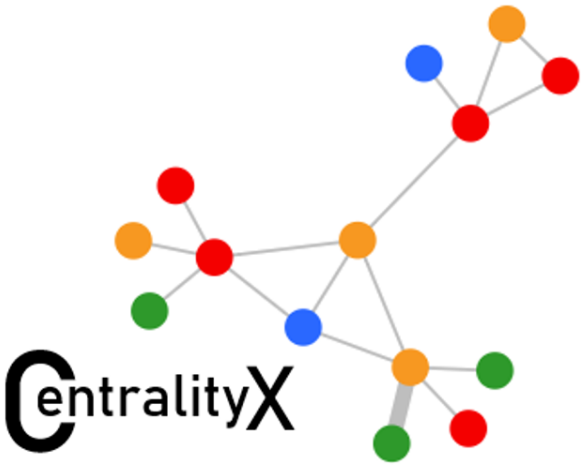
CentralityX



A D3 Network Analysis Extension with Centrality Measure Scores.
Built on D3.js V.4 with JSNetworkX (Javascript implementation of NetworkX).

Connects Entities | Reveals Relationships

QlikTech AU Presales Initiative 2019
Author: Anthony Garbin

Download:

CentralityX.zip

CentralityX Tutorial.qvf

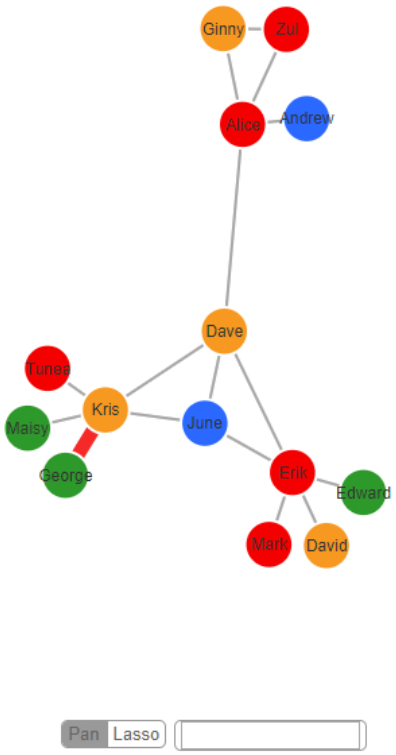
Description:

Centrality Measures are well established network analysis algorithms that tell us how about the relationships in a network.
A Network is simply a group of Nodes associated together.
Nodes are simply entities, similar to Nouns in the English Language
Nodes are connected together via Edges.
Edges are simply the relationships between Nodes, similar to Verbs in the English Language.

Why?:

Network Diagrams with Centrality Measures can quickly show us all the Entities and how they Entities are associated to other Entities together with the association relationship, the importance of each Entity and the strength of the relationships between entities.

What does it look like?



Please explain:

In the diagram above each circle is a node and each line is an edge. Each node can be selected directly to show direct associations, filtered by the lasso tool or filtered via dimension outside of the extension.

Data Model:

The data model can be as simple as two fields in a table. For example

edges:

LOAD * INLINE [

nodeA,nodeB

Dave,Anne

Dave,Robert

Robert,Beth

];

The extension requires at least two Dimensions, in this case we would assign "nodeA" and "nodeB"

Does it matter which name appears on which side? Yes and No.

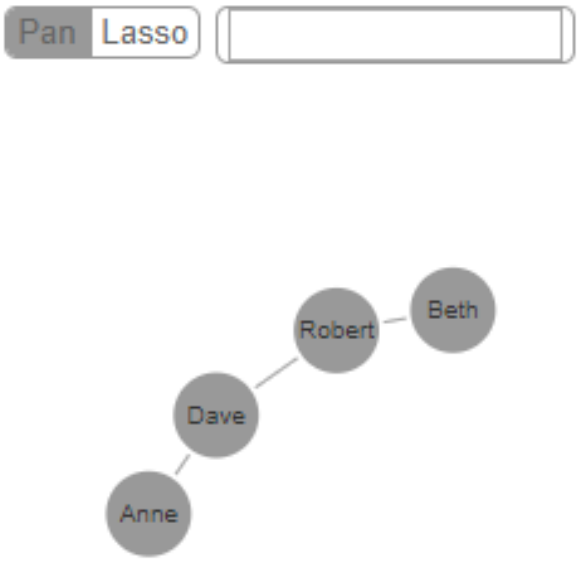
If nodeA only contains people from one set and nodeB only contains people from another set (and people are not populated on both sides), we call this is an "Exclusive Set". This is a simple data model and causes no confusion when using the extension.

If nodeA and nodeB contain people from the same set (and people are populated on both sides), we call this a "Non-Exclusive Set". There are implications when using this extension. The problem here is that when a person is on both sides, the extension network diagram

will show incomplete relationships when the user starts to apply filters. This is because the filters are only looking at the person on the one side that was filtered. It ignores the same person if they are also on the other side. This is mitigated by using

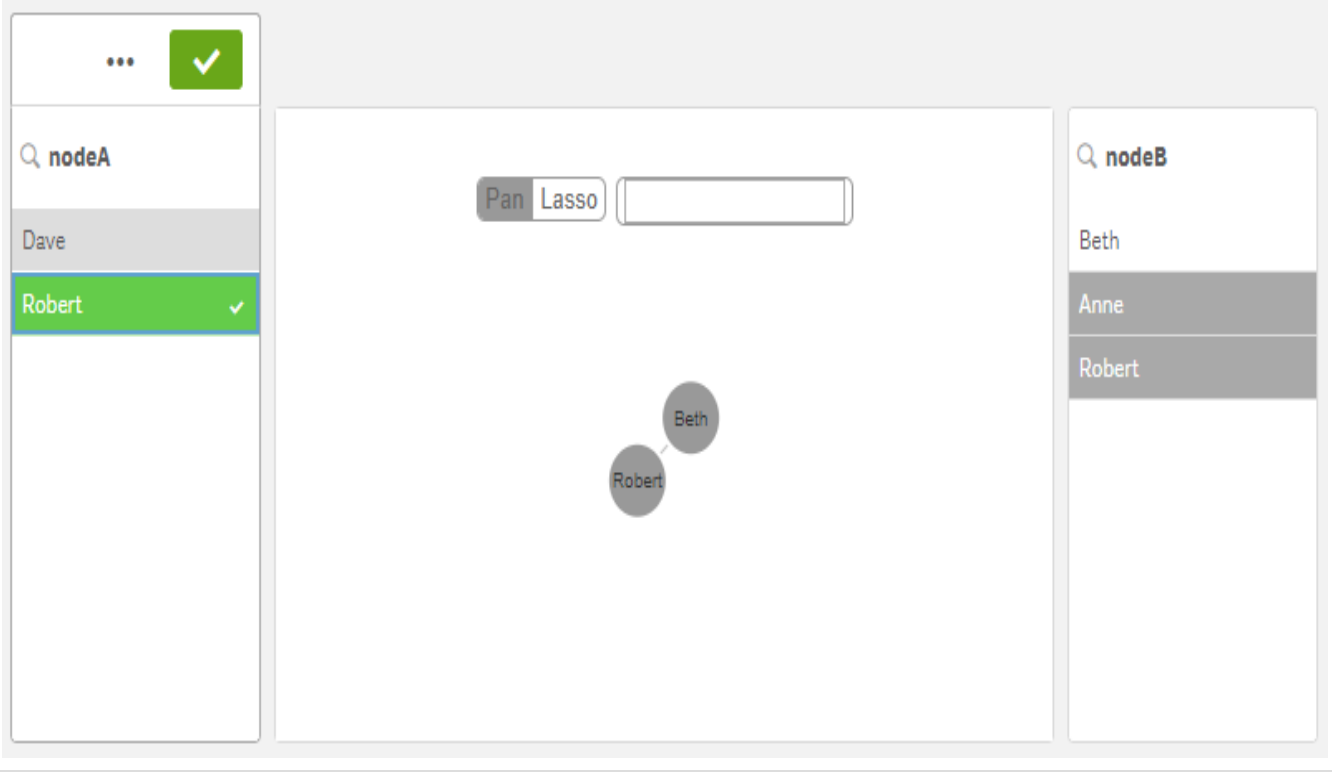
"Reverse Duplication" explained further in this article.

This simple table shows us that Robert has a relationship with both Dave and Beth.



However look what happens when you start to filter.

Non-Exclusive Set - when filtered (notice how people are either on one or the other, notice how filtering for Robert is not complete)



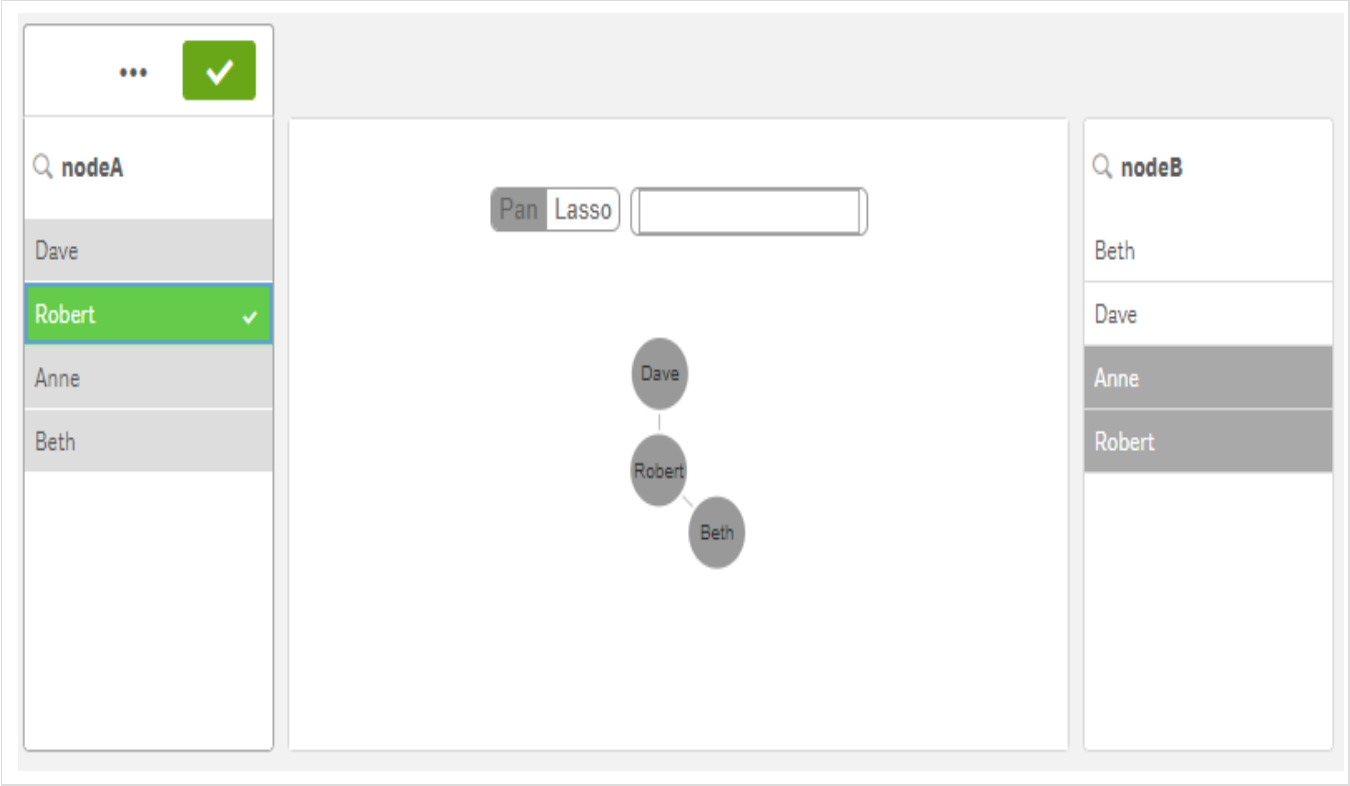
Reverse Duplication:

This is achieved by taking a copy of the edges and reversing the sides. This allow us to filter a person on either side and return back all relationships. The extension only allow each person to show once, therefore we won't see duplication in the diagram.

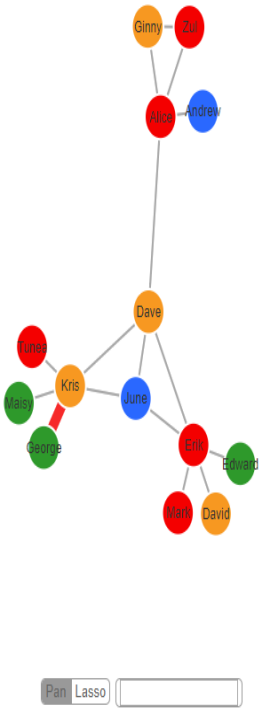
```
edges:
LOAD * INLINE [
nodeA,nodeB
Dave,Anne
Dave,Robert
Robert,Beth
];
NOCONCATENATE
edges_duplicated:
LOAD
   nodeA,
   nodeB
RESIDENT edges;

LOAD
   nodeA as nodeB,
   nodeB as nodeA
RESIDENT edges;
DROP TABLE edges;
```

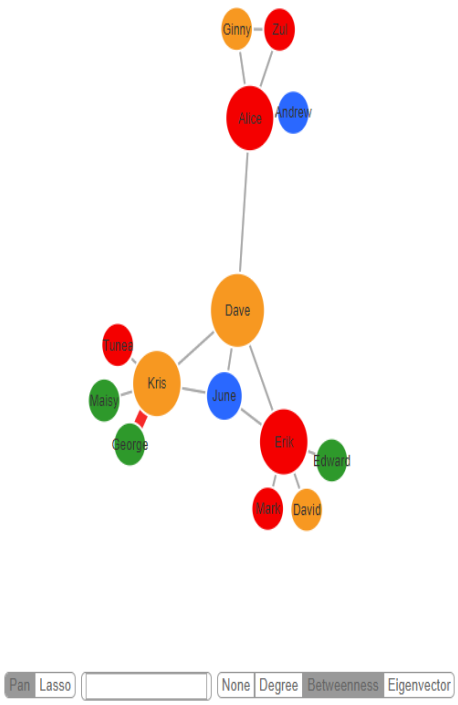
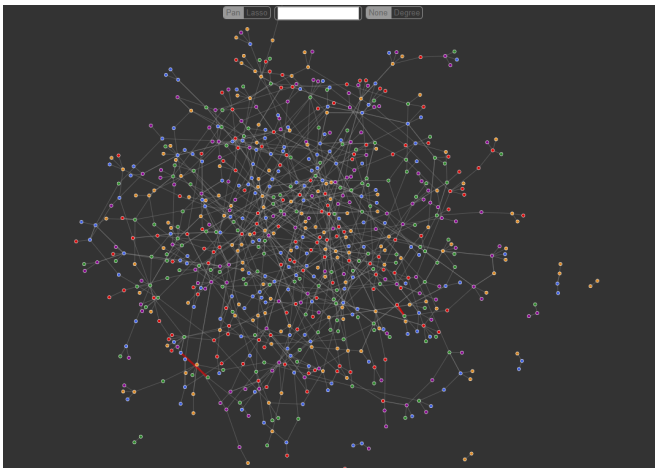
In this version we see that Robert has a relationship with both Dave and Beth.

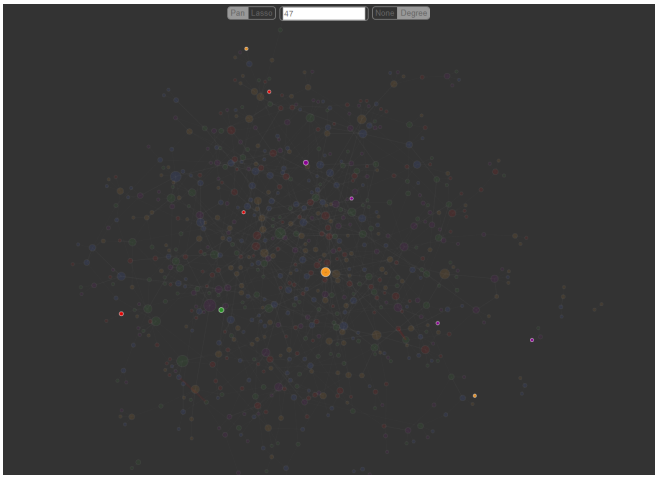
Non-Exclusive Set Reverse Duplicated - when filtered (Notice that we can select a person from either side and filtering show complete network!)



More advanced graphs:

Description	Screenshot
14 Nodes with Edge Weight and Color. Notice the relationship between Kris and George. We used an edge weight column to make the edge thicker and we used to a simple expression to change the color to Red when the edge weight is more than 1.	

14 Nodes with Node Size set to Betweenness Score. Dave has the highest Betweenness Score which is understandable when looking his position in the network. A betweenness analogy: Imagine you have a network of friends and there is a party but you were not invited! Who would you approach in the network to find out where the party is?	
1000 Random Nodes 2000 Random Relationships 5 Random Groups with Colors Edge Weight Thickness and Color. Even with this mess, can you spot several strong relationships indicated by strong red edges? Can you spot several small networks that don't seem to be associated with the main blob? Can you spot any concentration of interlinked groups by looking for clusters of colors? *No, nothing that is worth mentioning, plus this is random data, so probability was low!*	

Same as above + Search for any node with '47'. Notice how we can easily find visual clues for node 47. We can see that there are 10 other nodes with similar names, we can see that 47 is part of the main group, we can see that 47 belongs to the 'Orange' group and we can see that 47 has a large number of directly associated nodes due the the degree sizing of the node.	

Spaghetti Ball Problem:

The problem with Network Diagrams (like most network charts) is that they get too messy and difficult to read when the node size goes up! There are solutions to this issue which will be explained below along with some Load Script Exercises. Essentially we can either create logical groups with our nodes so that there are less nodes to display. One approach is to create Communities so that we have a way to group up the nodes, for example we can group nodes by the location or region that the node resides. After seeing relationships between two locations we could then start to look at individual nodes in those two locations. Another approach is to use what I call Strongest Neighbor Algorithm. A Neighbor is a node that is directly associated with another node. By examining the number of neighbors of every node and comparing which directly associated node has the most neighbors we can establish a new hierarchy such that the strongest neighbor of any cluster will proxy (represent) all it's weaker neighbors in the network diagram. The algorithm allows for the all the main players to be linked together, reducing network size however still providing lots of important information for the end user. All weaker neighbor are merged together, stored in the proxy node description and then searched for later.

TBC

Requests for New Features:

Yes, absolutely I will take requests for new features that don't break the primary design vision.

Early questions and requests:-

Item	Request
1	Does it work with QSE Smart Search?
2	Can it display Associate Index Colors (Green/White/Grey) when filtering?
3	Can it display advanced node details on node selection?
4	Can you change the node shape to represent different node attributes?
5	Can you add images to the nodes?
6	Can you expand/collapse nodes to show further associates?

Properties:

Panel	Item	Property	Description	Requirements	Comments
Dimensions	Dimension Field		Requires two entities linked together by a relationship	Numeric or Text Min 2 Dimensions Max 2 Dimensions	Using integers for node ID can boost performance.

Dimensions	Dimension Field	Node Label	Either the node ID or another alternative name. Displayed as label.	Numeric or Text	If omitted node ID is used as Label.
		Node Description	Further description of the node. Only displayed on mouse over.	Number or Text	null if omitted
		Node Color	Each node can be colored coded.	Hex '#ffffff'	Can be set via Expression Language
Measures Fields	Measure Field		Can add optional 2 measures to describe the edges.	Optional	
		Edge_Weight	Describes the size of the relationship to show frequency or strength. The relationship line will be thicker or thinner. Edge_Weights values are displayed on mouse over.	Optional	Can be set via Expression Language. Use "Map To" to assign correctly.
		Edge_Color	Describes the color of the relationship to show frequency or strength. For example the edge may be more "Red" when the relationship is stronger.	Optional	Can be set via Expression Language
Appearance	Nodes	Default Node Color	Override the default Node Color. Overridden by Node Color set in Dimension.	Optional	
		Initial Node Size Min	Override the initial min size of each node. Helps adjust the visibility of nodes for either small or large networks.	Optional	
		Initial Node Size Max	Override the initial max size of each node. Helps adjust the visibility of nodes for either small or large networks.	Optional	
		Node Searching	Change what part of the node is included in the node searching feature. Can be Node Name, Node Description or Both.	Required	
	Edges	Default Edge Color	Override the default Edge Color. Overridden by Edge Color set in Measures.	Optional	
		Min Edge Width (pixels)	Override the initial min thickness of each node. Helps adjust the visibility of edges for either small or large networks.	Optional	
		Max Edge Width (pixels)	Override the initial max thickness of each node. Helps adjust the visibility of edges for either small or large networks.	Optional	
	Labels	Label Size	Override the initial Label Font Size.	Optional	

		Label Font	Override the initial Label Font.	Optional	
		Label Color	Override the initial Label Font Color.	Optional	
		Show on Zoom Ratio	Labels can appear when the diagram is in zoom presentation mode. This helps to clean up large diagrams so that Labels only appear when the user has zoomed in a sufficient way.	Required	Default 0, means always show Label
	Options	Degree	Centrality Measure. Calculates the number of directly associated nodes to ever node. The size of the node will increase to indicate a higher Degree.	Optional	Performance gains if not used.
		Betweenness	Centrality Measure. Calculates how central a node is in the connection of other nodes.The size of the node will increase to indicate a higher Betweenness score.	Optional	Performance gains if not used.
		Eigenvector	Centrality Measure. Calculates how influential a node is by looking at the number of important surrounding nodes. The size of the node will increase to indicate a higher Eigenvector score.	Optional	Performance gains if not used.
		AlphaDecay (0.00 - 1.00)	D3 setting. Used to configure how long D3 will continue to perfect the Force Directed Layout. Each object repels itself away from other objects. This takes time to calculate.	Required	0.00 is perfect completion, 1.00 disables movement. 0.05 is a good compromise.
		Force Directed Strength	D3 setting. Used in calculating how nodes repel from each other.	Required	0 = no override.
		Forece Distance Max	D3 setting. Used in calculating how far away nodes should try to position themselves away from other nodes.	Required	0 = no override
		Menu Position X	Positions the Extension menu either Left, Center or Right.	Required	
		Menu Position Y	Positions the Extension menu either Top or Bottom.	Required	
		Node Count Warning	A value in which the user is prompted to Continue processing. App designers can elect to set this value or leave as zero.	Required	0 = no warning

		Color Theme	Select either Light or Dark theme. Only changes the Extension background, all other label and menu items are unchanged.If using Dark, consider changing Label to lighter colors.	Required	

Features:

Feature	Description	Comments
Pan	Network diagram can be panned left/right/up/down.	
Zoom	Mouse scrolling allows the Zoom In and Zoom Out.	
Lasso	Nodes can be selected inside the extension using the Lasso tool.	
Node Drag	Nodes can be dragged to adjust the placement.	
Node Click	Nodes can be clicked once to reveal the directly associated nodes. Other nodes fade into background.	
Node Hover	Node descriptions can be revealed when hovering over Node.	
Node Search	Node Names and Descriptions can be searched and highlighted using the Search tool.	Use options to change Search behavior
Node Color	Node Color can be used to describe a Node attribute such as community, group, type etc..	
Edge Hover	Edge weigh values can be revealed when hovering over Edge.	
Edge Weight	Edges support weight to identify stronger or weaker relationships	
Edge Color	Edges support colors to identify stronger or weaker relationships. Edge colors can also be used to describe relationship attributes such as Like, Buys From etc..	
Performance Warning	A value can be used to warn the user that the number of nodes being processed is high and may be slow.	