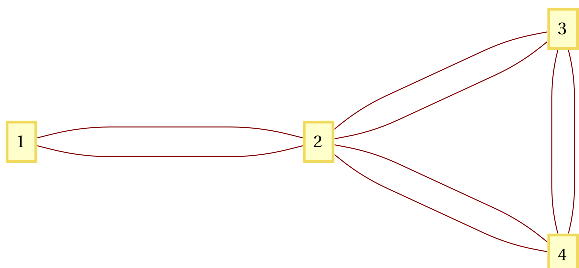
Graph Centrality Measures

The Example Graph

A undirected graph

In[159]:=
```
n = 4;
g = {1 -> 2, 2 → 1, 2 -> 3, 2→4, 3 -> 4, 3 -> 2, 4 → 2, 4 → 3};
A = AdjacencyMatrix[g];
GraphPlot[g, VertexLabeling -> True]
```

Out[162]=



Degree Centrality

Degree centrality is just the in-degree of each node.

In[163]:=
```
DC = A.{1,1,1,1};
MatrixForm[DC];
```

$$\begin{pmatrix} 1 \\ 3 \\ 2 \\ 2 \end{pmatrix}$$

Normalized degree centrality means to divide the degree by the maximal possible degree, this allows to compare degree centrality between different graphs.

```
NDC = DC*1/(n-1);
MatrixForm[NDC];
```

$$\begin{pmatrix} \frac{1}{3} \\ 1 \\ \frac{2}{3} \\ \frac{2}{3} \end{pmatrix}$$

EigenVector Centrality

Eigenvector centrality is based on the idea that well connected nodes are central. Eigenvector centrality awards vertices scores proportional to the sum of the scores of its neighbors. The formula is as follows.

$$x_v = \frac{1}{\lambda} \sum_{t \in M(v)} x_t = \frac{1}{\lambda} \sum_{t \in G} a_{v,t}\, x_t$$

Where $x_v$is the centrality score for vertex v and λ is the eigenvalue where the corresponding eigenvector is be non-negative, which implies that λ is the largest eigenvalue among all eigenvalues for the adjacency-matrix. $a_{v,t}$is a element of adjacency matrix and is 1 if a edge exist between vertex v and t. In vector notation the equation above can be written as follows.

$$\mathrm{Ax} = \lambda \mathrm{x}$$

To compute the Eigenvector centrality you can use the power method to find the dominant eigenvalue.

In[338]:=

```
EC = EigenvectorCentrality[g];
MatrixForm[EC];
```

$$\begin{pmatrix} 0.145362 \\ 0.315449 \\ 0.269594 \\ 0.269594 \end{pmatrix}$$

Computing it by hand:

```
lambda = Eigenvalues[A, 1];
x = Eigenvectors[A,1][[1]];
EC2 = Normalize[N[x]]*1/lambda[[1]]; (* 1/lambda is a normalization *)
MatrixForm[EC2];
```

$$\begin{pmatrix} 0.129877 \\ 0.281845 \\ 0.240876 \\ 0.240876 \end{pmatrix}$$

Closeness Centrality

Closeness centrality measures the mean distance from a vertex to any other vertex. This means that a vertex that is central geographically in the graph will be a high closeness centrality.

```
CC = ClosenessCentrality[g];
MatrixForm[CC];
```

Out[349]= 2

$$\begin{pmatrix} 0.6 \\ 1. \\ 0.75 \\ 0.75 \end{pmatrix}$$

Compute it by hand.

In[404]:=
```
d1 = {Length[FindShortestPath[g,1,2]],Length[FindShortestPath[g,1,3]],Length[FindShortestI
d2 = {Length[FindShortestPath[g,2,1]],Length[FindShortestPath[g,2,3]],Length[FindShortestI
d3 = {Length[FindShortestPath[g,3,2]],Length[FindShortestPath[g,3,1]],Length[FindShortestI
d4 = {Length[FindShortestPath[g,4,2]],Length[FindShortestPath[g,4,3]],Length[FindShortestI
CC2 = {1/Total[d1],1/Total[d2],1/Total[d3],1/Total[d4]};
MatrixForm[CC2];
```

$$\begin{pmatrix} \frac{1}{8} \\ \frac{1}{6} \\ \frac{1}{7} \\ \frac{1}{7} \end{pmatrix}$$

Closeness centrality can be normalized by multiplying with (n-1)

In[413]:=
```
CC3 = (n-1)*CC2;
MatrixForm[CC3]
```

$$\begin{pmatrix} \frac{3}{8} \\ \frac{1}{2} \\ \frac{3}{7} \\ \frac{3}{7} \end{pmatrix}$$

However this normalization wont work for disconnected graph since then a distance of ∞ will ruin the computation. We can remedy this with harmonic closeness centrality.

In[426]:=
```
d1 = {1/Length[FindShortestPath[g,1,2]],1/Length[FindShortestPath[g,1,3]],1/Length[FindSh
d2 = {1/Length[FindShortestPath[g,2,1]],1/Length[FindShortestPath[g,2,3]],1/Length[FindSh
d3 = {1/Length[FindShortestPath[g,3,2]],1/Length[FindShortestPath[g,3,1]],1/Length[FindSh
d4 = {1/Length[FindShortestPath[g,4,2]],1/Length[FindShortestPath[g,4,3]],1/Length[FindSh
CC4 = {Total[d1],Total[d2],Total[d3],Total[d4]};
MatrixForm[CC4];
```

$$\begin{pmatrix} \frac{7}{6} \\ \frac{3}{2} \\ \frac{4}{3} \\ \frac{4}{3} \end{pmatrix}$$

Betweenness Centrality

Betweenness centrality measures for every node how many shortest paths it is included in. The intuition for this measure is that bridge-nodes that connect components together is important for the overall graph and will get high betweeness centrality.

```
BC = BetweennessCentrality[g];
MatrixForm[BC];
```

$$\begin{pmatrix} 0. \\ 4. \\ 0. \\ 0. \end{pmatrix}$$

I.e all shortest paths go through node number 2.

KendallTau Rank

Kendall Tau Rank can be used to compare how the difference centrality measures are in concordance with each other or if they disagree with each other. The formula for KendallTau rank is:

$$\tau = \frac{n_c - n_d}{n(n-1)/2}$$

Where τ is the Kendall tau rank, n_c is the number of concordant pairs and n_d is the number of discordant pairs. Perfect agreement when tau=1, complete disagreement with tau = -1. If A is ranked over B in both rankings then it is a concordant pair. If C is ranked over D in some ranking and D is ranked over C in another ranking it is a discordant pair.

In[490]:=
```
KendallTau[NDC, EC]
```
Out[490]= 0.912871

In[491]:=
```
KendallTau[NDC, CC]
```
Out[491]= 1.

In[492]:=
```
KendallTau[NDC, BC]
```
Out[492]= 0.774597

In[493]:=
```
KendallTau[EC, CC]
```
Out[493]= 0.912871

In[494]:=
```
KendallTau[EC, BC]
```
Out[494]= 0.707107

In[495]:=
```
KendallTau[CC, BC]
```
Out[495]= 0.774597

Kim Hammar

kimham@kth.se
14/10-2017