Module 3: Graph Theory and Centrality Measures

Miranda Myers
9-9-18

① Adjacency Matrix

A	1	2	3	4	5
1	0	1	1	0	1
2	1	0	0	1	1
3	1	0	0	1	0
4	0	1	1	0	0
5	1	1	0	0	0

B	1	2	3	4	5	6
1	0	0	0	1	0	1
2	0	0	1	1	1	1
3	0	1	0	1	0	0
4	1	1	1	0	0	1
5	0	1	0	0	0	1
6	1	1	0	1	1	0

② Degree Centralities

A:

	sum	/ n-1	degree
1:	0 + 1 + 1 + 0 + 1 = 3	4	3/4
2:	1 + 0 + 0 + 1 + 1 = 3	4	3/4
3:	1 + 0 + 0 + 1 + 0 = 2	4	2/4
4:	0 + 1 + 1 + 0 + 0 = 2	4	2/4
5:	1 + 1 + 0 + 0 + 0 = 2	4	2/4

B:

	sum	/ n-1	degree
1:	0 + 0 + 0 + 1 + 0 + 1 = 2	5	2/5
2:	0 + 0 + 1 + 1 + 1 + 1 = 4	5	4/5
3:	0 + 1 + 0 + 1 + 0 + 0 = 2	5	2/5
4:	1 + 1 + 1 + 0 + 0 + 1 = 4	5	4/5
5:	0 + 1 + 0 + 0 + 0 + 1 = 2	5	2/5
6:	1 + 1 + 0 + 1 + 1 + 0 = 4	5	4/5

③ Geodesics

A

from	to	geodesic
1	2	(1,2)
1	3	(1,3)
1	4	(1,3,4) or (1,2,4)
1	5	(1,5)
2	3	(2,1,3) or (2,4,3)
2	4	(2,4)
2	5	(2,5)
3	4	(3,4)
3	5	(3,1,5)
4	5	(4,2,5)

B

from	to	geodesic
1	2	(1,4,2) or (1,6,2)
1	3	(1,4,3)
1	4	(1,4)
1	5	(1,6,5)
1	6	(1,6)
2	3	(2,3)
2	4	(2,4)
2	5	(2,5)
2	6	(2,6)
3	4	(3,4)
3	5	(3,2,5)
3	6	(3,2,6) or (3,4,6)
4	5	(4,2,5) or (4,6,5)
4	6	(4,6)
5	6	(5,6)

④ Betweenness Centrality

1	2	3	4	5
0	0	0	0	0
0	0	0	0	0
0	.5	.5	0	0
0	0	0	0	0
.5	0	0	.5	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
1.5	1.5	.5	.5	0 ← sum
$\frac{1.5}{6}$	$\frac{1.5}{6}$	$\frac{.5}{6}$	$\frac{.5}{6}$	0 ← betweenness

Denominator = $\frac{(5-1)(5-2)}{2} = 6$

1	2	3	4	5	6
0	0	0	.5	0	.5
0	0	0	1	0	0
0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	1	0	0	0	0
0	.5	0	.5	0	0
0	.5	0	0	0	.5
0	0	0	0	0	0
0	0	0	0	0	0
0	2	0	2	0	2 ← sum
0	$\frac{2}{10}$	0	$\frac{2}{10}$	0	$\frac{2}{10}$ ← betweenness

Denominator = $\frac{(6-1)(6-2)}{2} = 10$

⑤ Closeness

A	1	2	3	4	5	sum	n-1	closeness
1	-	1	1	2	1	5	4	4/5
2	1	-	2	1	1	5	4	4/5
3	1	2	-	1	2	6	4	4/6
4	2	1	1	-	2	6	4	4/6
5	1	1	2	2	-	6	4	4/6

B	1	2	3	4	5	6	sum	n-1	closeness
1	-	2	2	1	2	1	8	5	5/8
2	2	-	1	1	1	1	6	5	5/6
3	2	1	-	1	2	2	8	5	5/8
4	1	1	1	-	2	1	5	5	5/6
5	2	1	2	2	-	1	7	5	5/8
6	1	1	2	1	1	-	6	5	5/6

⑥ Agents with influence

Network A

Agents that may have greater influence: 1 and 2, because they have the greatest values for degree, betweenness, and centrality.

Network B

Agents that may have greater influence: 2, 4, and 6 for the same reason as above.

Degree = opportunity to directly influence

Betweenness = informal power

Closeness = indirect influence

7) Density

A: 5 choose 2 $= \frac{5!}{2!3!} = 10$ possible links in network

There are $\frac{6}{10}$ edges $= 60\%$

B: 6 choose 2 $= \frac{6!}{2!4!} = 15$ possible links

There are $\frac{9}{15}$ edges $= 60\%$

8) Diameter

The longest geodesics for networks A and B, calculated in #3, are both 2, so diameter = 2.

Source Code

```
from igraph import *
import math


def createGraphA():
    g = Graph()
    g.add_vertices(5)
    g.add_edges([(0,1), (0,2), (0,4), (3,1), (3,2), (4,1)])
    return g


def createGraphB():
    g = Graph()
    g.add_vertices(6)
    g.add_edges([(0, 5), (0, 3), (1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (3, 5), (4,
            5)])
    return g


def getMaxAgents(list):
    m = max(list)
    return [i + 1 for i, j in enumerate(list) if j == m]


def calculateDegreeCentralities(g, numVertices, file):
    degreeCentralities = [x /(numVertices - 1.0) for x in g.degree()]
    file.write('Degree Centrality: \n\t' + str(degreeCentralities))
    file.write('\nAgents with Max Degree Centrality: \n\t' +
            str(getMaxAgents(degreeCentralities)) + '\n\n')


def createAdjacencyMatrix(g, file):
    file.write('Adjacency Matrix: \n')
    g.write_adjacency(file)


def getBetweenness(g, numVertices):
    denominator = ((numVertices - 1.0) * (numVertices - 2.0)) / 2.0
    return [round(x / denominator, 3) for x in g.betweenness()]


def calculateBetweenness(g, numVertices, file):
    betweenness = getBetweenness(g, numVertices)
    file.write('Betweenness: \n\t' + str(betweenness))
    file.write('\nAgents with Max Betweenness: \n\t' +
            str(getMaxAgents(betweenness)) + '\n\n')


def calculateCloseness(g, file):
    closeness = [round(x, 3) for x in g.closeness()]
    file.write('Closeness: \n\t' + str(closeness))
    file.write('\nAgents with Max Closeness: \n\t' +
            str(getMaxAgents(closeness)) + '\n\n')


def calculageDensity(g, file):
    file.write('Density: \n\t' + str(g.density()) + '\n\n')
```

```
def calculageDiameter(g, file):
    file.write('Diameter: \n\t' + str(g.diameter()) + '\n\n')


def writeResults(g, file):
    numVertices = g.vcount()
    calculateDegreeCentralities(g, numVertices, file)
    calculateBetweenness(g, numVertices, file)
    calculateCloseness(g, file)
    calculageDensity(g, file)
    calculageDiameter(g, file)
    createAdjacencyMatrix(g, file)


def setAgentNumbers(g):
    agents = []
    for i in range(1, g.vcount() + 1):
        agents.append(i)

    g.vs['agent'] = agents


def setLabels(g):
    numVertices = g.vcount()
    betweenness = getBetweenness(g, numVertices)
    maxAgents = getMaxAgents(betweenness)
    labels = []
    for i in range(1, g.vcount() + 1):
        if i in maxAgents:
            labels.append('agent-' + str(i))
        else:
            labels.append(' ')

    g.vs['label'] = labels


def setColors(g):
    colorDict = {True: 'purple', False: 'teal'}
    g.vs['color'] = [colorDict[agentNumber % 2 == 0] for agentNumber in
            g.vs['agent']]


def setVertexSize(g):
    numVertices = g.vcount()
    g.vs['size'] = [x * 200 for x in getBetweenness(g, numVertices)]


def nCr(n, r):
    f = math.factorial
    return f(n) / f(r) / f(n - r)


def displayNetwork(g):
    setLabels(g)
    setAgentNumbers(g)
    setColors(g)
    setVertexSize(g)
    plot(g)
```

```
def main():
    fileA = open('centrality-measures-graph-a.txt', 'w')
    a = createGraphA()
    writeResults(a, fileA)

    fileB = open('centrality-measures-graph-b.txt', 'w')
    b = createGraphB()
    writeResults(b, fileB)

    displayNetwork(b)
    displayNetwork(a)


main()
```

Text Output:

Graph A:
Degree Centrality:
[0.75, 0.75, 0.5, 0.5, 0.5]
Agents with Max Degree Centrality:
[1, 2]

Betweenness:
[0.25, 0.25, 0.083, 0.083, 0.0]
Agents with Max Betweenness:
[1, 2]

Closeness:
[0.8, 0.8, 0.667, 0.667, 0.667]
Agents with Max Closeness:
[1, 2]

Density:
0.6

Diameter:
2

Adjacency Matrix:
0 1 1 0 1
1 0 0 1 1
1 0 0 1 0
0 1 1 0 0
1 1 0 0 0

Graph B:
Degree Centrality:
[0.4, 0.8, 0.4, 0.8, 0.4, 0.8]
Agents with Max Degree Centrality:
[2, 4, 6]

Betweenness:
[0.0, 0.2, 0.0, 0.2, 0.0, 0.2]
Agents with Max Betweenness:
[2, 4, 6]

Closeness:
[0.625, 0.833, 0.625, 0.833, 0.625, 0.833]
Agents with Max Closeness:
[2, 4, 6]

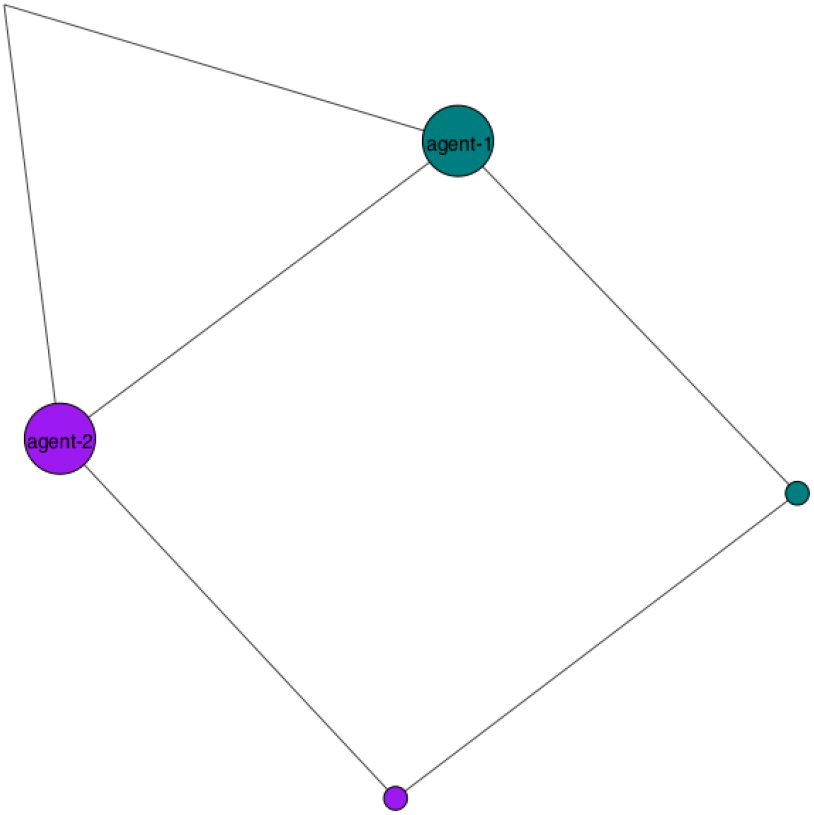
Density:
0.6

Diameter:
2

Adjacency Matrix:
0 0 0 1 0 1
0 0 1 1 1 1
0 1 0 1 0 0
1 1 1 0 0 1
0 1 0 0 0 1
1 1 0 1 1 0

Images

Graph A:



Graph B: