The Engineering World #DataScience 28 & 29

May 31, 2018

AKKAL BAHADUR BIST
DATA SCIENTIST AT
KATHMANDU INSTITUTE OF APPLIED SCIENCES (KIAS)
Center for Conservation Biology (CCB)

1 DIRECTED NETWORK ANALYSIS

1.1 Simulating Social Network(Directed Network Analysis)

```
In [1]: import numpy as np
        import pandas as pd
        from pylab import rcParams
        import seaborn as sb
        import matplotlib.pyplot as plt
        import networkx as nx
```

```
In [2]: %matplotlib inline
        rcParams ['figure.figsize'] = 5,4
        sb.set_style ('whitegrid')
```

1.1.1 Generating a grapg object and edgelist

```
In [3]: DG = nx.gn_graph(11, seed=25)
        for line in nx.generate_edgelist(DG, data=False):
            print (line)
```

```
1 0
2 0
3 2
4 3
5 0
6 4
7 3
8 0
9 8
10 1
```

```
In [4]: print (DG.node[0])
```

```
{}
```

```
In [5]: print (DG.node[5])
```

```
{}
```

1.1.2 Assigning attributes to nodes

```
In [6]: DG.node[0]['name']='Alice'
```

```
In [7]: print (DG.node[0])
```

```
{'name': 'Alice'}
```

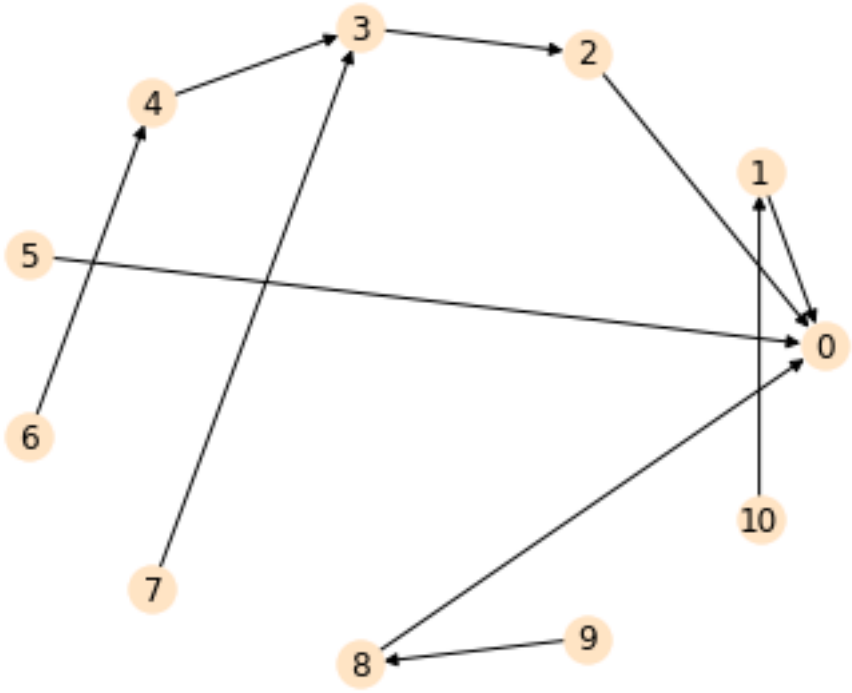
```
In [8]: DG.node[0]['name']='Alice'
        DG.node[1]['name']='Akkal'
        DG.node[2]['name']='Janak'
        DG.node[3]['name']='Laxman'
        DG.node[4]['name']='Bikash'
        DG.node[5]['name']='Dinesh'
        DG.node[6]['name']='Amin'
        DG.node[7]['name']='Sunil'
        DG.node[8]['name']='Kiran'
        DG.node[9]['name']='Surya'
```

```
In [9]: DG.add_nodes_from([(0,{'age':25}),(1,{'age':31}),(2,{'age':18}),(3,{'age':47}),(4,{'age'
        print(DG.node[1])
```

```
{'name': 'Akkal', 'age': 31}
```

```
In [10]: DG.node[0]['name']='M'
         DG.node[1]['name']='M'
         DG.node[2]['name']='F'
         DG.node[3]['name']='M'
         DG.node[4]['name']='F'
         DG.node[5]['name']='F'
         DG.node[6]['name']='M'
         DG.node[7]['name']='F'
         DG.node[8]['name']='M'
         DG.node[9]['name']='F'
```

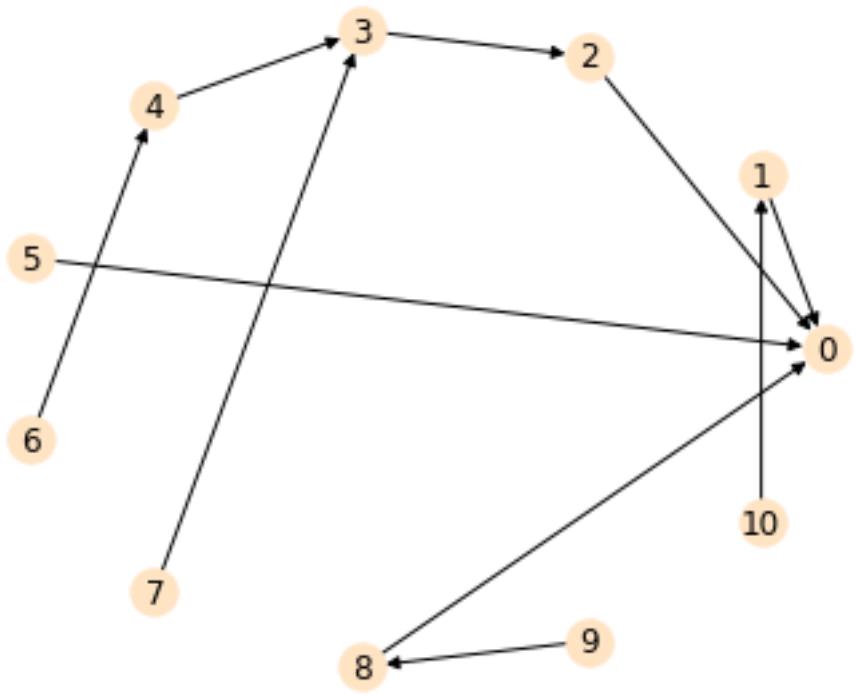
```
In [11]: nx.draw_circular(DG, node_color = 'bisque', with_labels = True)
```



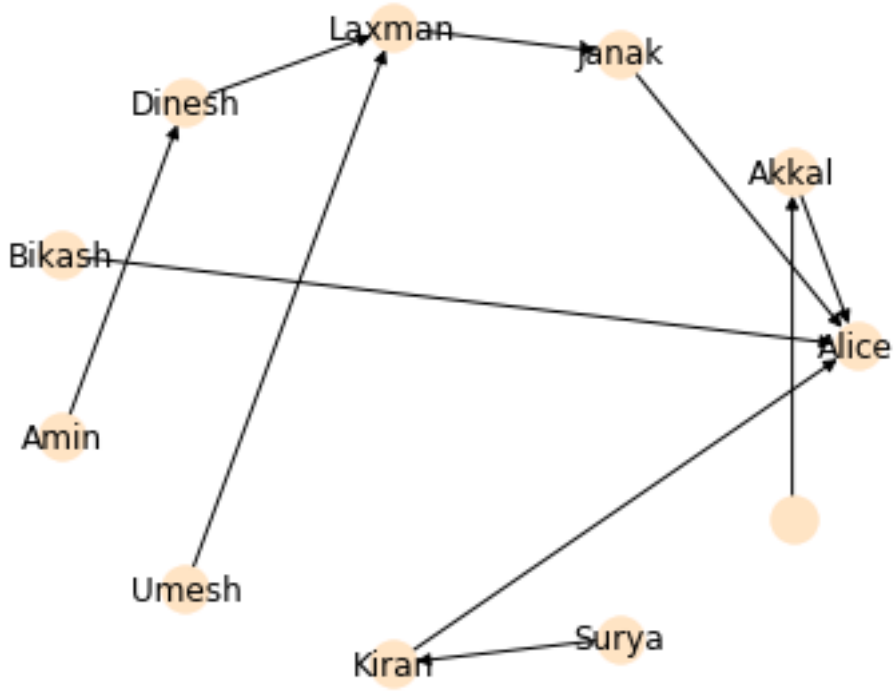
1.1.3 Visualize your network graph

```
In [12]: labeldict = {0:'Alice', 1:'Akkal', 2:'Janak', 3:'Laxman', 4:'Dinesh', 5:'Bikash', 6:'Am
```

```
In [13]: nx.draw_circular(DG, node_color = 'bisque', with_labels = True)
```

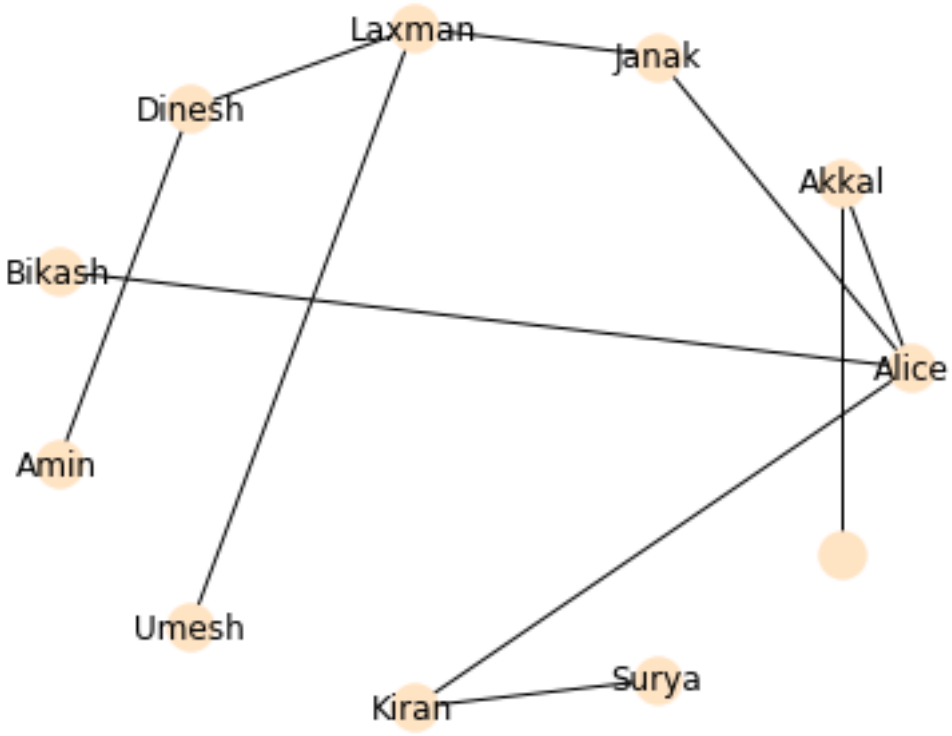



```
In [14]: nx.draw_circular(DG, labels = labeldict, node_color = 'bisque', with_labels = True) #ad
```




```
In [15]: G = DG.to_undirected()

In [16]: nx.draw_circular(G, labels = labeldict, node_color = 'bisque', with_labels = True)
```



2 NETWRK ANALYSIS GRAPH INSPECTION AND STATES ON NODES

2.0.1 Analyzing a Social Network

```
In [17]: DG = nx.gn_graph(7, seed=25)
         for line in nx.generate_edgelist(DG, data=False):
             print(line)

         DG.node[0]['name']='Alice'
         DG.node[1]['name']='Akkal'
         DG.node[2]['name']='Janak'
         DG.node[3]['name']='Laxman'
         DG.node[4]['name']='Bikash'
         DG.node[5]['name']='Dinesh'
         DG.node[6]['name']='Amin'
```

```
1 0
2 0
3 2
```

```
4 3
5 0
6 4
```

```
In [18]: G = DG.to_undirected()
```

```
In [19]: print (nx.info(DG))
```

```
Name:
Type: DiGraph
Number of nodes: 7
Number of edges: 6
Average in degree:   0.8571
Average out degree:   0.8571
```

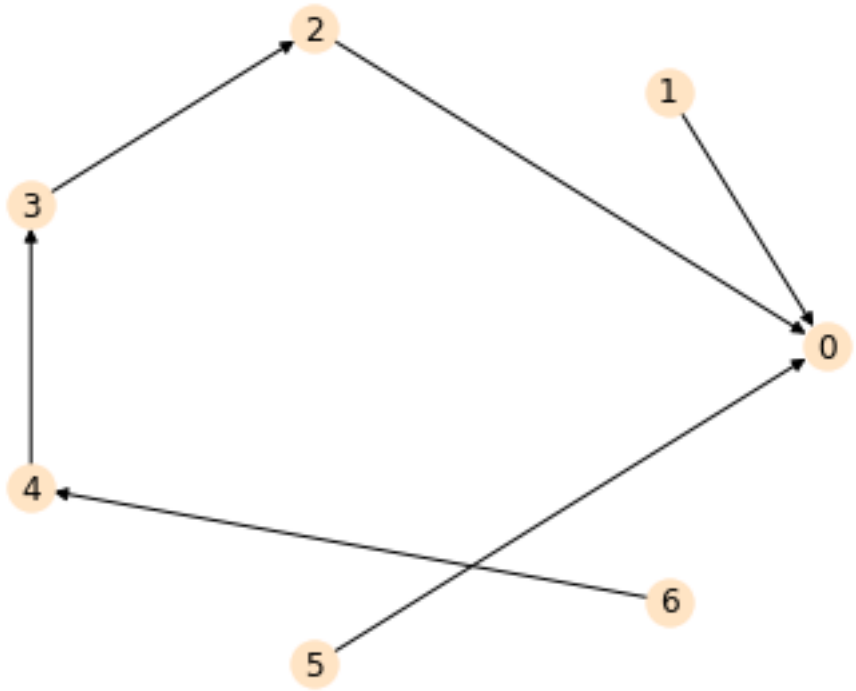
Considering degree in a social network

```
In [20]: DG.degree()
```

```
Out[20]: DiDegreeView({0: 3, 1: 1, 2: 2, 3: 2, 4: 2, 5: 1, 6: 1})
```

Identifying successor nodes

```
In [21]: nx.draw_circular(DG, node_color = 'bisque', with_labels = True) #add labels in graph pl
```



```
In [22]: DG.successors(3)

Out[22]: <dict_keyiterator at 0x7f77d5bf62c8>

In [23]: DG.neighbors(4)

Out[23]: <dict_keyiterator at 0x7f77d5be4318>

In [24]: G.neighbors(4)

Out[24]: <dict_keyiterator at 0x7f77d5be4b38>
```