NETWORK ANALYSIS IN PYTHON II

Introduction to the dataset

Dataset & case study introduction

- College forum posting dataset, 6 months
- Node partitions: students, forums
- Activities in the chapter:
  - Constructing a graph from a pandas DataFrame
  - Computing unipartite projections of a bipartite graph
  - Visualization
  - Time series filtering & analysis
- Recap previously used functions

Graphs from DataFrames

```
In [1]: df
Out[1]:
   customers    products
0  customerA    product1
1  customerB    product2
...

In [2]: G = nx.Graph()

In [3]: G.add_nodes_from(df['products'], bipartite='products')

In [4]: G.add_nodes_from(df['customers'], bipartite='customers')

In [5]: G.nodes()
Out[5]: ['product1', 'customerC', 'product2', 'customerB', 'customerA']

In [6]: G.edges()
Out[6]: []
```

Graphs from DataFrames

```
In [7]: G.add_edges_from(zip(df['customers'], df['products']))

In [8]: G.edges()
Out[8]: [('product1', 'customerC'), ('product1', 'customerA'),
   ...: ('customerC', 'product2'), ('product2', 'customerB')]
```

Bipartite projections

```
In [9]: cust_nodes = [n for n in G.nodes() if G.node[n]
   ...:               ['bipartite'] == 'customers']

In [10]: prod_nodes = [n for n in G.nodes() if G.node[n]
    ...:                ['bipartite'] == 'products']

In [11]: prodG = nx.bipartite.projected_graph(G, nodes=prod_nodes)

In [12]: custG = nx.bipartite.projected_graph(G, nodes=cust_nodes)

In [13]: prodG.nodes()
Out[13]: ['product1', 'product2']

In [14]: custG.nodes()
Out[14]: ['customerC', 'customerB', 'customerA']
```

NETWORK ANALYSIS IN PYTHON II

Let’s practice!

NETWORK ANALYSIS IN PYTHON II

Time based filtering

Key concepts

- Filtering graphs
- Datetime
- Visualization

Filtering edges

```
In [1]: G.edges(data=True)[0:5]
Out[1]:
[(0, 17, {'sale_count': 1}),
 (0, 18, {'sale_count': 1}),
 (0, 19, {'sale_count': 2}),
 (0, 12, {'sale_count': 14}),
 (0, 13, {'sale_count': 9})]

In [2]: [(u, v) for u, v, d in G.edges(data=True) if d['sale_count'] >= 10]
Out[2]: [(0, 12), (1, 19), (5, 16), (6, 13), (7, 17), (7, 19), (8, 18)]
```

Datetime

```
In [1]: from datetime import datetime, timedelta

In [2]: year = 2011

In [3]: month = 11

In [4]: day1 = 10

In [5]: day2 = 6

In [6]: date1 = datetime(year, month, day1)

In [7]: date2 = datetime(year, month, day2)

In [8]: date1 > date2
Out[8]: True
```

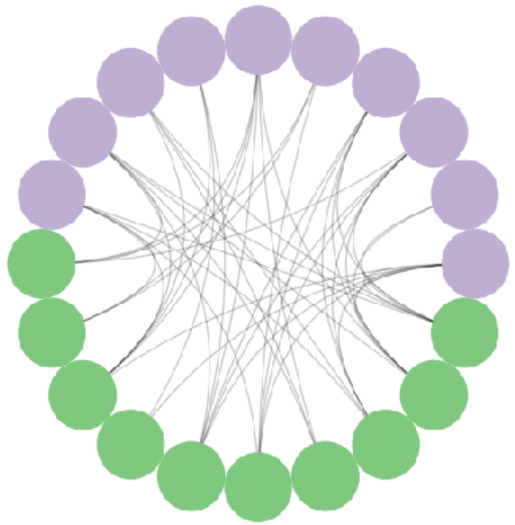
Graph visualization

```
In [1]: from nxviz import CircosPlot

In [2]: c = CircosPlot(G, node_grouping='bipartite',
   ...:                node_color='bipartite')

In [3]: c.draw()

In [4]: plt.show()
```



NETWORK ANALYSIS IN PYTHON II

Let’s practice!

NETWORK ANALYSIS IN PYTHON II

Time series analysis

Time series

- Global vs. local analysis
- Analyze evolving graph statistics
- Make plots of key evolving stats

Datetime arithmetic

```
In [1]: date1
Out[1]: datetime.datetime(2011, 11, 10, 0, 0)

In [2]: days = 4

In [3]: td = timedelta(days)

In [4]: date1 + td
Out[4]: datetime.datetime(2011, 11, 14, 0, 0)
```

Degree centrality

```
In [5]: G
Out[5]: <networkx.classes.graph.Graph at 0x10e7c04a8>

In [6]: nx.degree_centrality(G)
Out[6]: {1: 0.5, 'c': 0.5, 'b': 0.25, 2: 0.5, 'a': 0.25}

In [7]: nx.bipartite.degree_centrality(G, [1, 2])
Out[7]: {1: 0.6666666666666666, 2: 0.6666666666666666, 'b': 0.5,
   ...:  'c': 1.0, 'a': 0.5}
```

NETWORK ANALYSIS IN PYTHON II

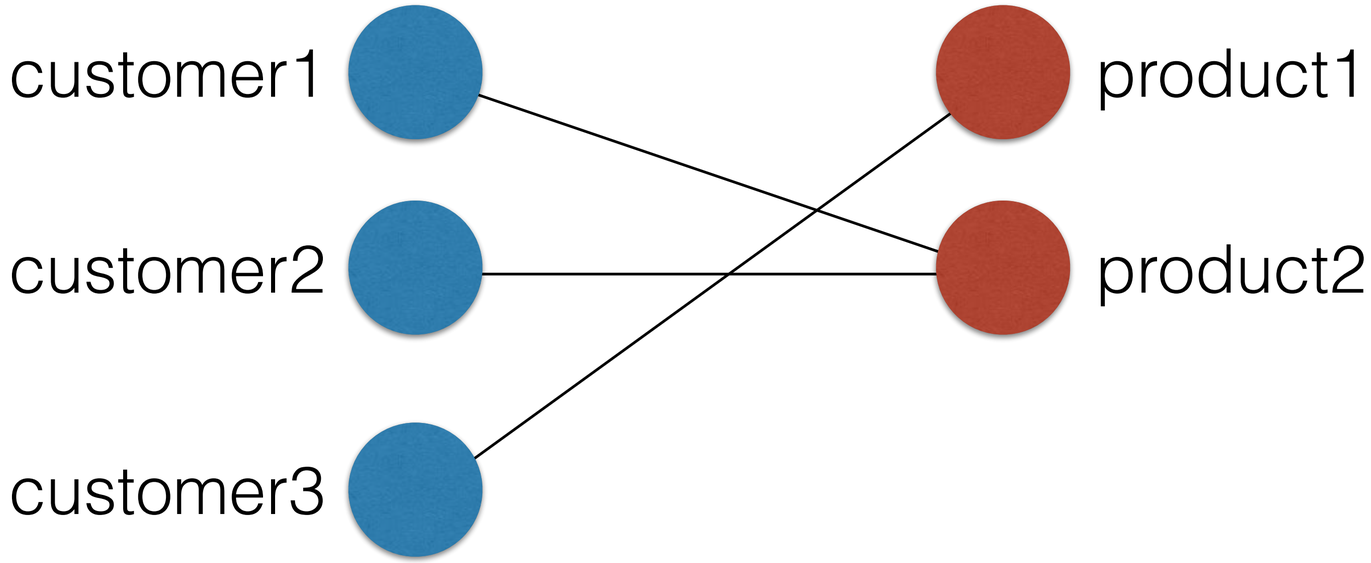
Let’s practice!

NETWORK ANALYSIS IN PYTHON II

Congratulations!

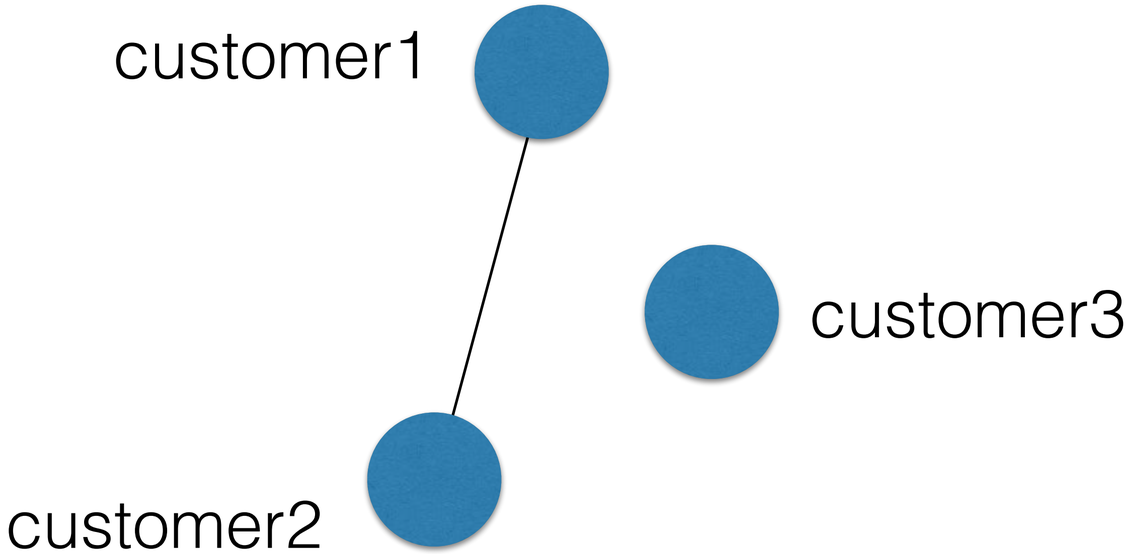
What you've learned

- Bipartite graphs



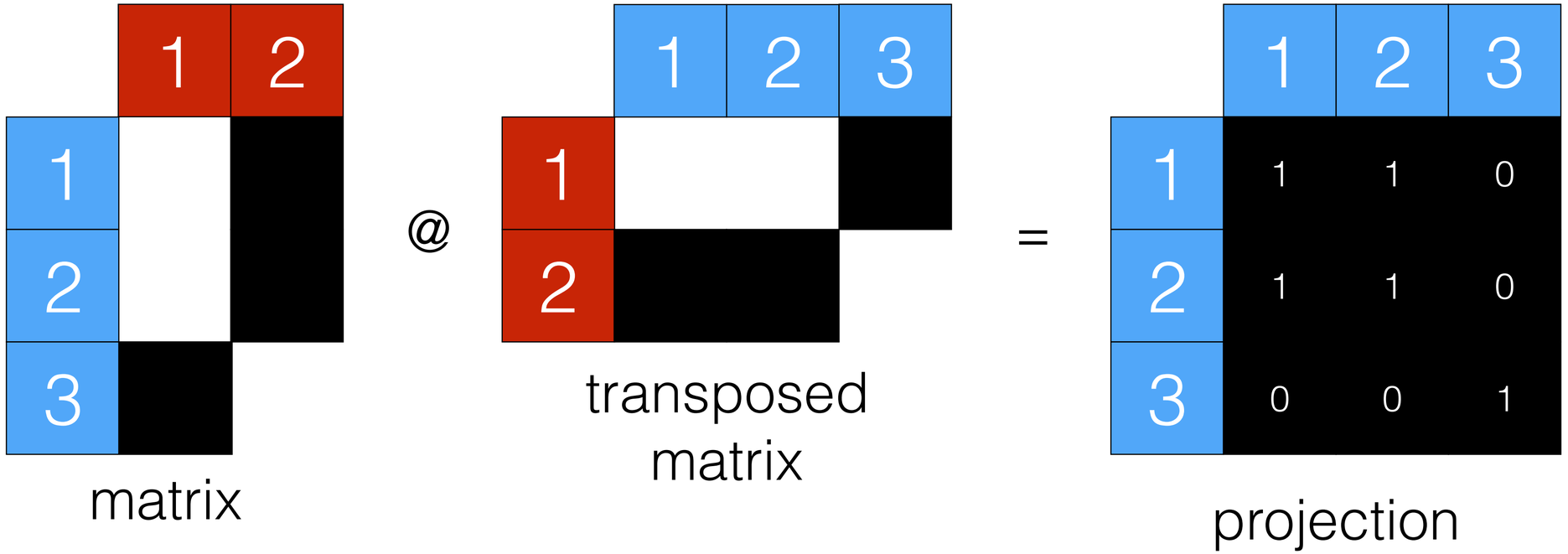
What you've learned

- Bipartite graphs
- Projections



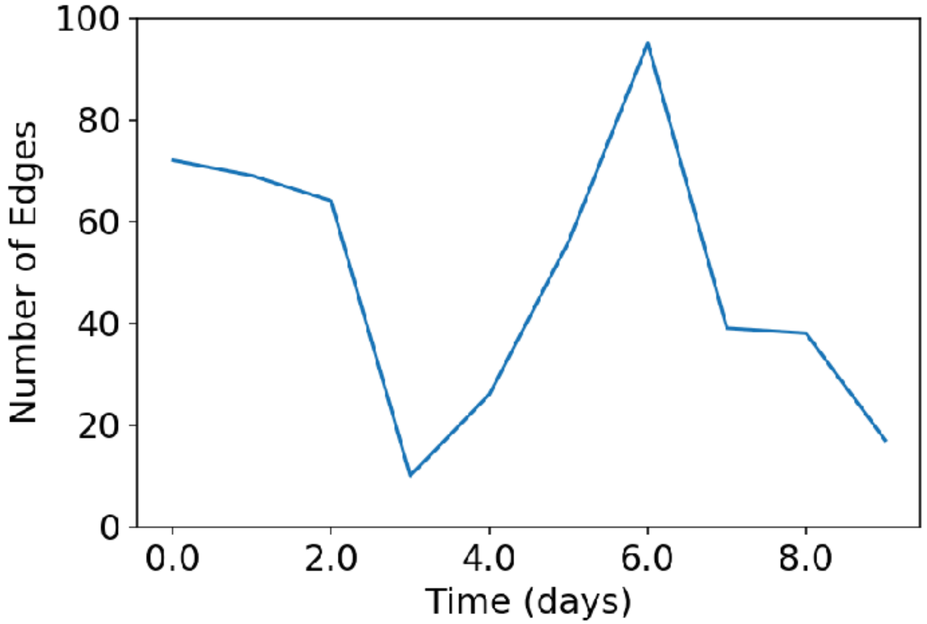
What you've learned

- Bipartite graphs
- Projections
- Matrix representation



What you've learned

- Bipartite graphs
- Projections
- Matrix representation
- Time series



NETWORK ANALYSIS IN PYTHON II

Congratulations!