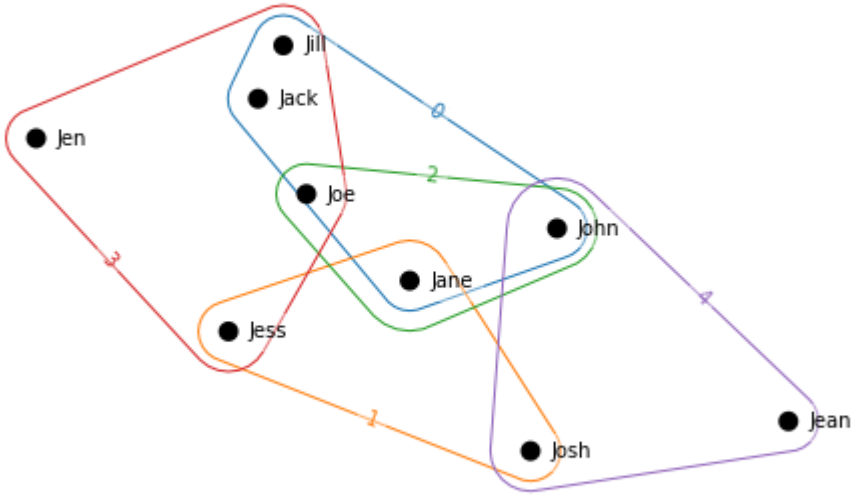
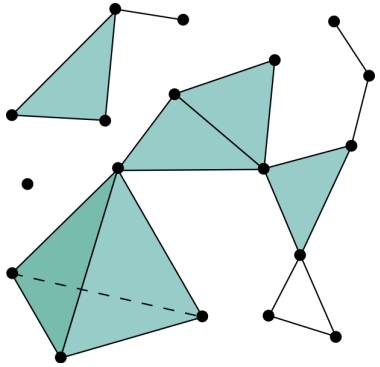
Higher-Order Interactions in Networks

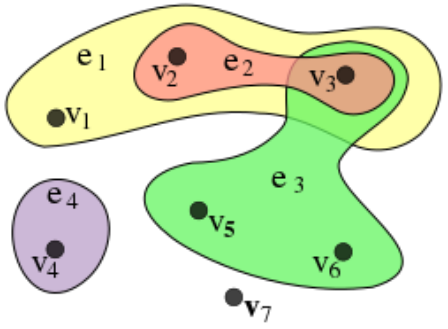


Hiroki Sayama
sayama@binghamton.edu

Higher-order interactions

- Interactions among more than two nodes (i.e., relationships that cannot be captured by simple pairwise edges)
- Typical modeling frameworks:
  - Hypergraphs
  - Simplicial complexes





Images from Wikipedia

Heads up (2)

- This is even much newer than temporal or multilayer network research!!
- Tools for higher-order interaction modeling and analysis are very limited
- Can be even more math-intense than temporal or multilayer network research

Examples

- Categorical relationships among objects (e.g., memberships in sets)
- Communications among multiple participants (e.g., group discussions)
- Multiple-species interactions in ecosystems

Hypergraphs

Hypergraph

- A generalization of a graph in which a "hyperedge" can join any number of nodes
  - i.e., a hyperedge is a set of nodes
  - It is no longer dots and lines!

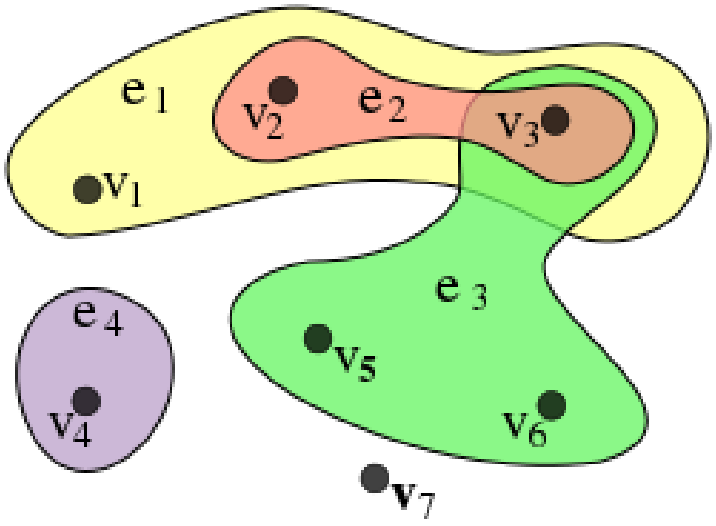
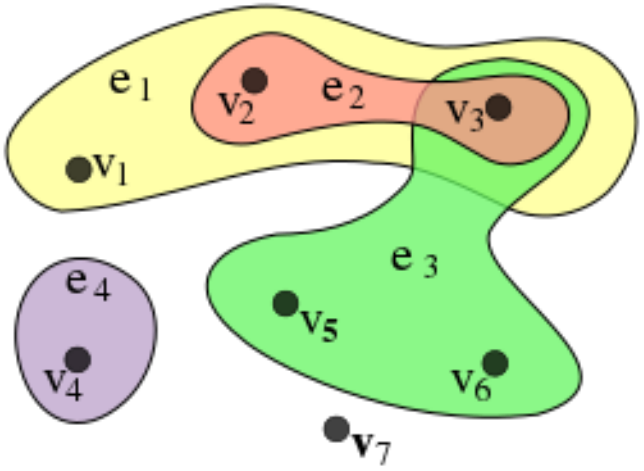


Image from Wikipedia

Matrix representation

- Hypergraph can be represented by a rectangular incidence matrix



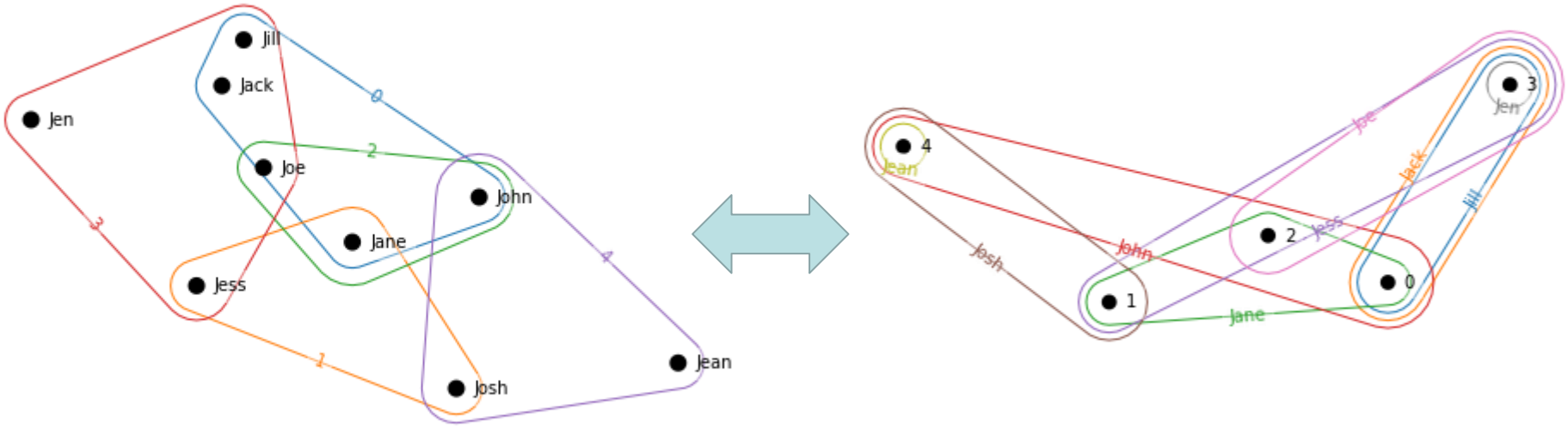
edges

nodes

1	0	0	0
1	1	0	0
1	1	1	0
0	0	0	1
0	0	1	0
0	0	1	0
0	0	0	0

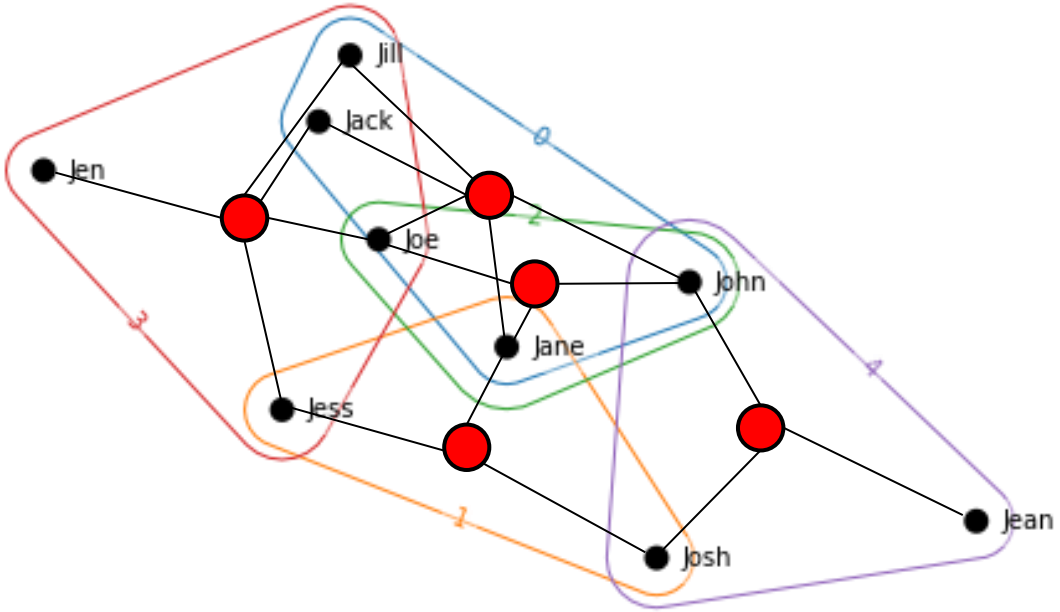
Duality of hypergraphs

- Every hypergraph has its "dual" obtained by swapping roles of nodes and edges (i.e., transposing its incidence matrix)



Representation as bipartite graphs

- Hypergraphs can be converted to ordinary graphs by creating new nodes to represent hyperedges

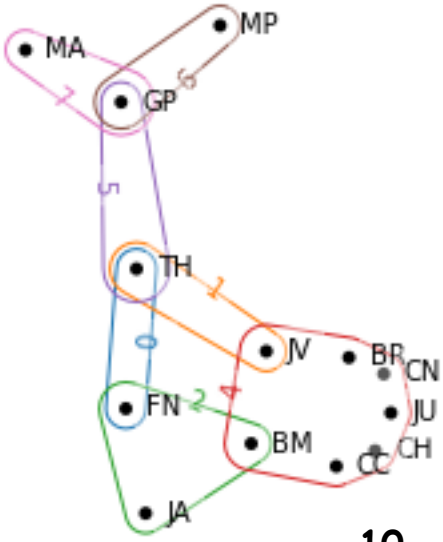


HyperNetX

- Developer team led by our Systems Science alumnus Cliff Joslyn! ☺

```
pip install hypernetx
import networkx as nx
import hypernetx as hnx
```

  - You may need to install additional packages too



HyperNetX basics (1)

- Hyperedges as dictionaries

```
g = hnx.Hypergraph({
    0: ('John', 'Jane', 'Jill', 'Jack'),
    1: ('Jess', 'Josh', 'Jane'),
    2: ('John', 'Jane', 'Joe'),
    3: ('Joe', 'Jill', 'Jen', 'Jess', 'Jack'),
    4: ('Jean', 'Josh', 'John')
})

hnx.draw(g)
```

HyperNetX basics (2)

- g.nodes
- g.edges
- g.shape
- g.incidence_dict
- g.incidence_matrix()
- g.degree(node)
- g.neighbors(node)
- g.size(edge)
- g.dual()
- g.connected_components()
- g.s_components(s)
- g.distance(i, j, s)
- g.edge_distance(k, l, s)

etc…

Connectedness in hypergraphs

- Connectivity can be measured at different levels of node sharing between hyperedges
  - Denoted by the "s" parameter in HyperNetX
    - E.g. "s = 2" means paths/distances are measured by assuming that only hyperedges that share 2 or more nodes are incidental to each other

Exercise

- Create and visualize a hypergraph model of your social surroundings by considering multiple groups (hyperedges) you belong to:
  - Research groups
  - Project teams
  - Family
  - Friend groups 1, 2, 3, …

    etc…

Exercise

- Analyze structural properties of the hypergraph model you just created
- Create and visualize its dual
- Who are connected from you with s=2, s=3, etc.?

Simplicial Complexes

Simplicial complex

- A more formal, structured way of representing multi-node interactions using combinations of "simplices" (plural of "simplex")
  - Edge: 1-simplex
  - Triangle: 2-simplex

    etc…

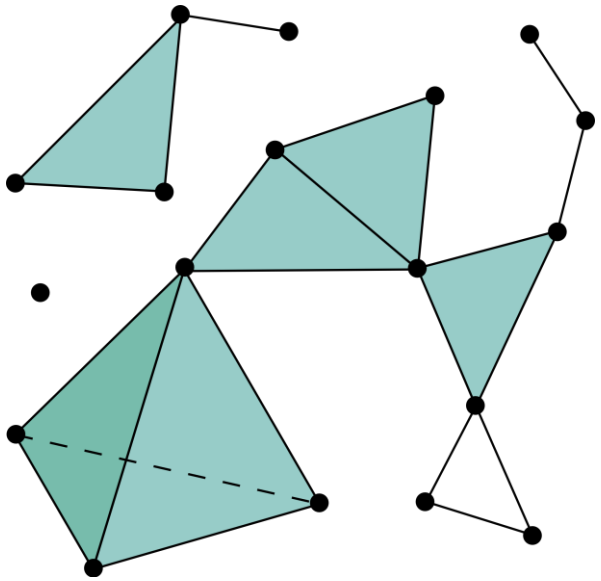
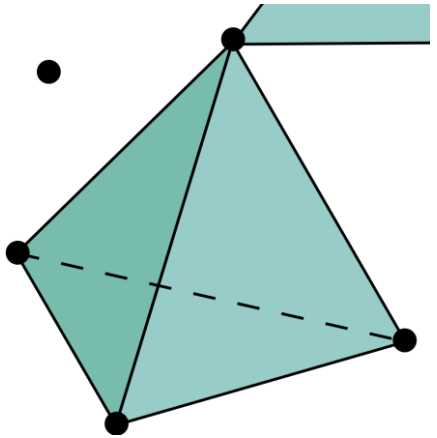


Image from Wikipedia

Properties of simplicial complexes

- Defined as a set of simplices
- All "subfaces" (i.e., lower-dimensional parts) of each simplex are also included in the simplicial complex
  - E.g. if a tetrahedron (3-simplex) is in a simplicial complex, then its four triangles and six edges must also be included in the set

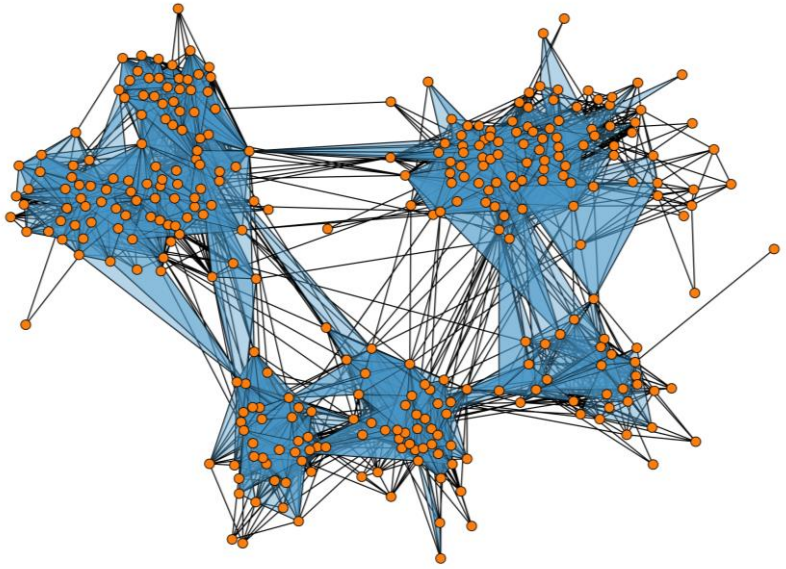


Hypergraph vs. simplicial complex?

- Hypergraphs are more open-ended and flexible than simplicial complexes
- Simplicial complexes assume any lower-dimensional interactions should also be present
  - A triad interaction means there must also be three dyad interactions too
  - Hypergraphs do not represent such assumptions

Simplicial complex in NetworkX

- You will need to do most of work yourself, but there are some tools available:
  - "py-draw-simplicial-complex" by Iacopo Iacopini https://github.com/iaciac/py-draw-simplicial-complex

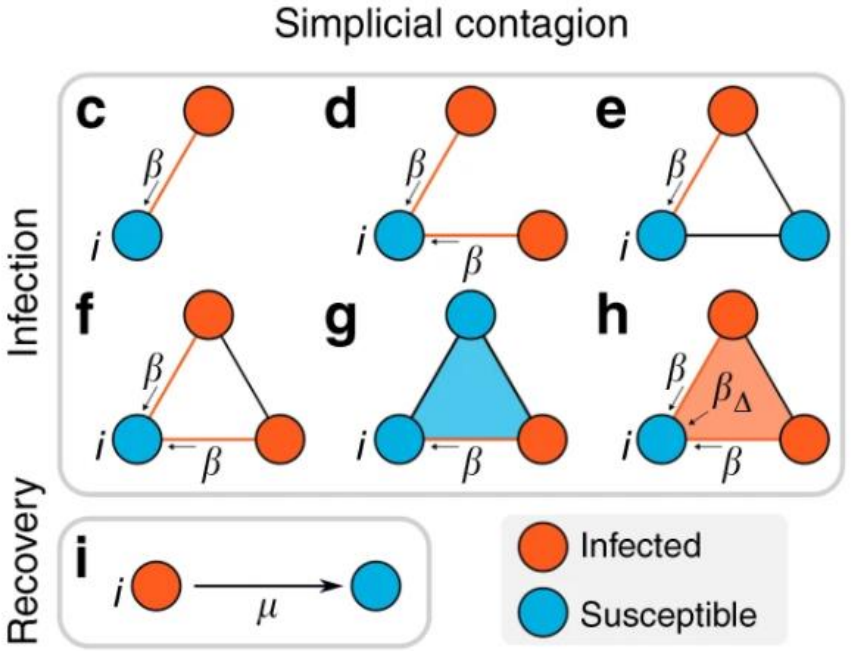


Exercise

- Rewrite the "social surroundings" hypergraph data you previously created into a simplicial complex (i.e., a set of simplices)
- Visualize it using py-draw-simplicial-complex

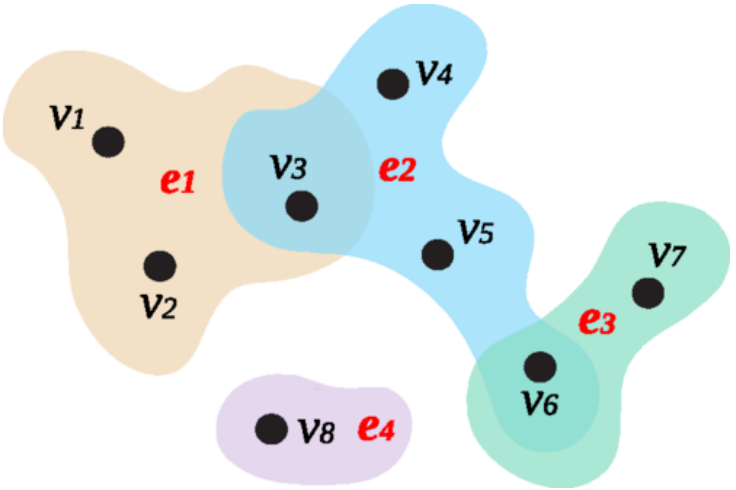
Application example (1): Dynamics on simplicial complexes

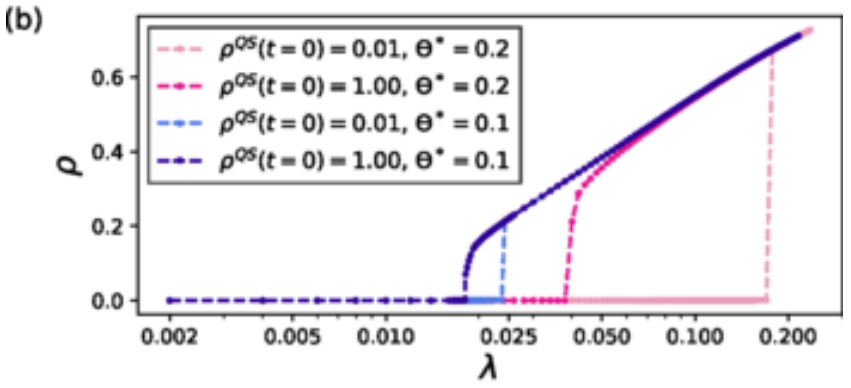
- Iacopini, I., et al. (2019) *Nature Communications, 10*(1), 1-9.
  - Contagion that takes into account the effects of triads shows an abrupt (explosive) jump of infection at a critical threshold



Application example (2): Dynamics on hypergraphs

- de Arruda, G. F., et al. (2020). *Physical Review Research, 2(2),* 023032.
  - Similar dynamical properties (e.g., bistability and explosive transition) are observed for hypergraphs





Warning

- Higher-order interactions in networks are still a VERY new topic in network science, so many concepts and methods are still being rapidly developed and tested
- Some of those may not survive

- This means you have a good chance to make big contributions to this area!!