Using Python for Hypergraph Learning

Focus on the CHESHIRE Algorithm

Moses Boudourides

Graduate Program on Data Science
School of Professional Studies
Northwestern University

Seminar at the Department of Sociology
University of Trieste, Italy
October 27, 2025

Northwestern | SCHOOL OF PROFESSIONAL STUDIES

Outline

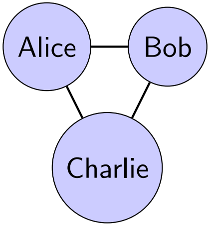
The Social Networks You Know

Simple Graphs:

- Nodes = People
- Edges = Friendships
- Perfect for pairwise relationships

Examples:

- Facebook friendships
- Twitter followers
- Email exchanges



Simple graph: pairwise connections

The Problem: Group Interactions

What about...

- A group chat with 5 friends?
- A research team working on a project?
- Students enrolled in the same course?
- A family gathering?

Challenge

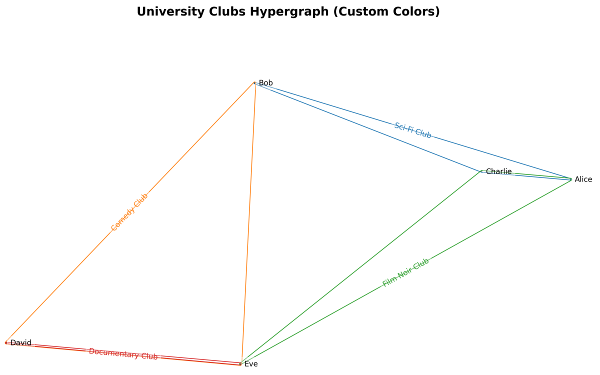
These are **group interactions** involving more than two people at once.
Simple graphs cannot naturally represent these relationships!

Enter Hypergraphs!

Hypergraph:

- **Nodes** = Entities
- **Hyperedges** = Groups
- Can connect any number of nodes

Feature	Graph	Hypergraph
Basic Unit	Edge	Hyperedge
Connections	2 nodes	2+ nodes
Example	Friendship	Group chat



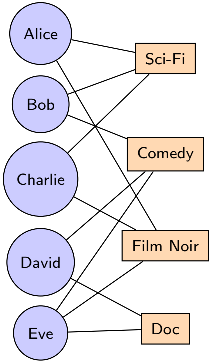
Example: University Clubs

Python Representation:

```
university_hypergraph = {
    'Sci-Fi Club':
        {'Alice', 'Bob', 'Charlie'
            },
    'Comedy Club':
        {'Bob', 'David', 'Eve'},
    'Film Noir Club':
        {'Alice', 'Charlie', 'Eve'
            },
    'Documentary Club':
        {'David', 'Eve'}
}
```

Key: Dictionary with hyperedges as keys, sets of nodes as values

Bipartite Visualization:

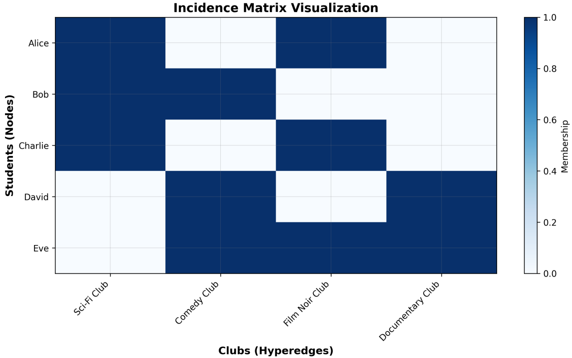


Mathematical Representation:

Incidence matrix H:

- **Rows** $=$ Nodes (students)
- **Columns** $=$ Hyperedges (clubs)
- $H[i,j] = 1$ if node i belongs to hyperedge j
- $H[i,j] = 0$ otherwise

This matrix is the foundation for machine learning on hypergraphs!



What is a Tensor?

Tensors are generalizations of familiar objects:

Dimension	Name	Example	Shape
0D	Scalar	5	()
1D	Vector	[1, 2, 3]	(3,)
2D	Matrix	Incidence matrix	(5, 4)
3D+	Tensor	Video (W $\times$ H $\times$ T)	(W, H, T)

Why Tensors Matter

- **Data** is represented as tensors
- **Neural networks** operate on tensors
- **GPUs** are optimized for tensor operations

Tensor Operations for Hypergraphs

Key Operations:

1. **Matrix Multiplication** (Message Passing):

```
# Node features to hyperedges
hyperedge_features = torch.matmul(H.T, X)

# Hyperedges back to nodes
X_updated = torch.matmul(H, hyperedge_features)
```

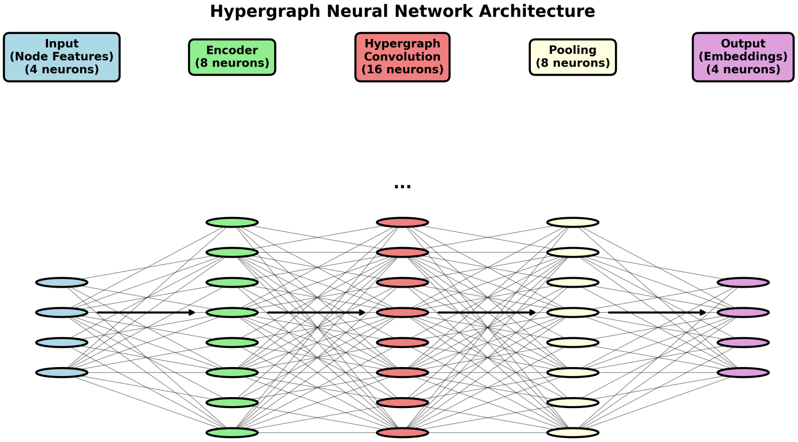
2. **Aggregation** (Computing degrees):

```
node_degrees = H.sum(dim=1)  # Sum across columns
hyperedge_sizes = H.sum(dim=0)  # Sum across rows
```

3. **Normalization**:

```
X_normalized = X / (node_degrees.unsqueeze(1) + 1e-8)
```

Neural Network Layers



Message Passing in Hypergraph Neural Networks

The Big Idea:

Nodes and hyperedges "talk" to each other:

1. **Nodes → Hyperedges**
   Each hyperedge aggregates information from its member nodes
2. **Hyperedges → Nodes**
   Each node receives information from all hyperedges it belongs to
3. **Learn**
   Through iteration, nodes learn meaningful representations

Introducing CHESHIRE

CHEbyshev Spectral HyperlInk pREdictor

A state-of-the-art deep learning method for hyperlink prediction

The Problem:

- Metabolic networks have missing reactions
- Experimental data is expensive
- Need computational predictions

CHESHIRE's Solution:

- Uses only network topology
- No experimental data needed
- Outputs confidence scores

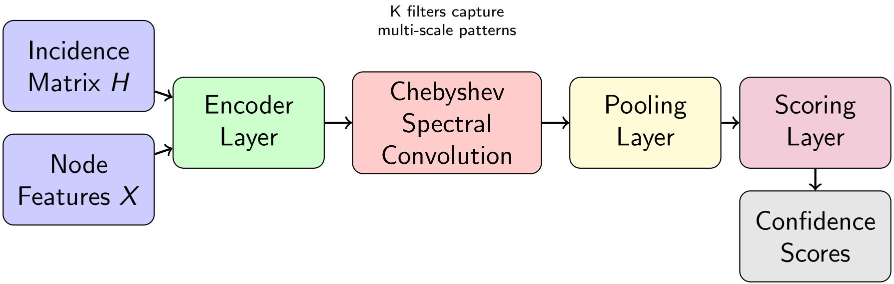
Key Features:

- Chebyshev spectral filters for efficient message passing
- Captures multi-hop relationships
- Scalable to large networks

Applications

- Metabolic network gap-filling
- Drug discovery
- Systems biology

CHESHIRE Architecture



Message passing through hypergraph structure $\rightarrow$

Why Chebyshev Spectral Convolution?

Traditional Convolutions:

- Work on grids (images)
- Fixed neighborhood structure
- Not suitable for graphs

Spectral Convolutions:

- Work on irregular structures
- Capture graph topology
- Mathematically principled

Chebyshev Polynomials:

- Make computation efficient
- Capture multi-hop relationships
- Avoid expensive eigendecomposition

Key Insight

Instead of just looking at immediate neighbors, Chebyshev filters consider neighbors of neighbors efficiently!

Multiple filters (K channels) capture different types of relationships

CHESHIRE Performance

Validation:

- Tested on 926 genome-scale metabolic models
- Outperforms other topology-based methods
- Improves phenotypic predictions

Impact:

- Accelerates metabolic model curation
- Reduces experimental costs
- Enables predictions for uncultivable organisms

Key Results:

- **High accuracy** in predicting missing reactions
- **Scalable** to large networks (1000s of nodes)
- **Interpretable** confidence scores
- **No experimental data** required

Publication

Chen et al. (2023). *Nature Communications*, 14(1), 2375.

Applications Beyond Biology

Social Sciences:

- Group dynamics analysis
- Community formation
- Collaboration prediction
- Social movement modeling

Computer Science:

- Recommendation systems
- Computer vision
- Natural language processing

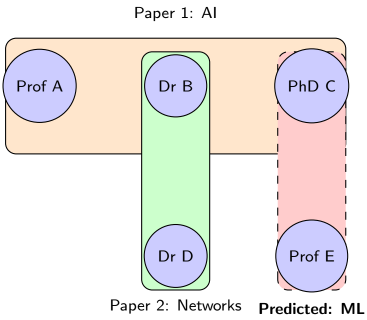
Other Domains:

- Political science (coalitions)
- Economics (multi-party transactions)
- Chemistry (reaction networks)
- Neuroscience (brain connectivity)

Key Insight

Any system with group interactions can benefit from hypergraph learning!

Predicting Future Collaborations



Hypergraph learning can predict which researchers are likely to collaborate next!

Key Takeaways

1. **Hypergraphs** naturally represent group interactions
2. **Tensors** are the data structures for machine learning
3. **Hypergraph Neural Networks** use message passing to learn from structure
4. **PyTorch** and PyTorch Geometric provide tools for implementation
5. **CHESHIRE** is a state-of-the-art algorithm for hyperlink prediction
6. These techniques apply to many disciplines, not just computer science

The key: Recognize when your data involves group interactions!

Resources

Papers:

- Chen et al. (2023). Teasing out missing reactions in genome-scale metabolic networks through hypergraph learning. *Nature Communications*, 14(1), 2375.
- Chen et al. (2024). A Survey on Hyperlink Prediction. *IEEE Transactions on Neural Networks and Learning Systems*, Volume: 35, Issue: 11, Page(s): 15034 - 15050.

Code & Libraries:

- CHESHIRE: `https://github.com/canc1993/cheshire-gapfilling`
- PyTorch Geometric: `https://pytorch-geometric.readthedocs.io/`
- HyperNetX: `https://github.com/pnnl/HyperNetX`

Tutorials:

- Hypergraph Neural Networks: `https://sites.google.com/view/hnn-tutorial`
- PyTorch: `https://pytorch.org/tutorials/`

Thank You!

Questions?

Contact: your.email@units.it