SAND2016-8016C

Advantages to Modeling Relational Data using Hypergraphs versus Graphs

Michael Wolf, Alicia Klinvex, and Daniel Dunlavy

IEEE HPEC

14 September, 2016

Sandia National Laboratories

Exceptional service in the national interest

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2016-4418 C.

Hypergraph Models for Data Science

Sandia National Laboratories

- Explore usage of hypergraphs for modeling complex, multiway (non-pairwise) relational data
- **Goal**: Improve data analysis – more accurate, faster
- **Questions we are trying to address**
  - Why should we use hypergraphs?
  - When are hypergraph models preferable over graph models?
  - What hypergraph model should be used?
  - How should we compute the solution to our hypergraph problems?
- **Specific problem**
  - Data clustering: Determine groupings of data objects given sets of relationships between/amongst objects

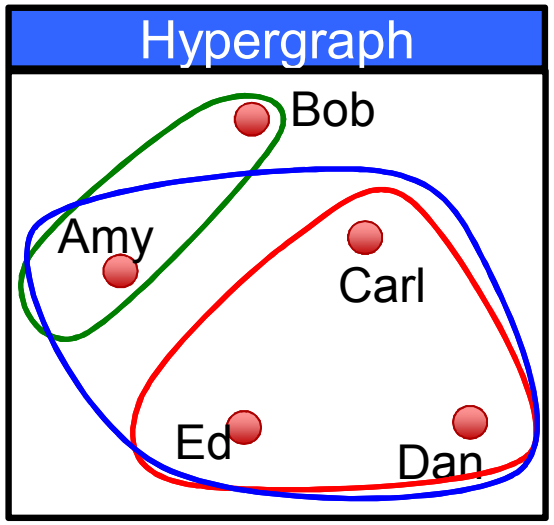
What is a hypergraph?

Hyperedges: Emails

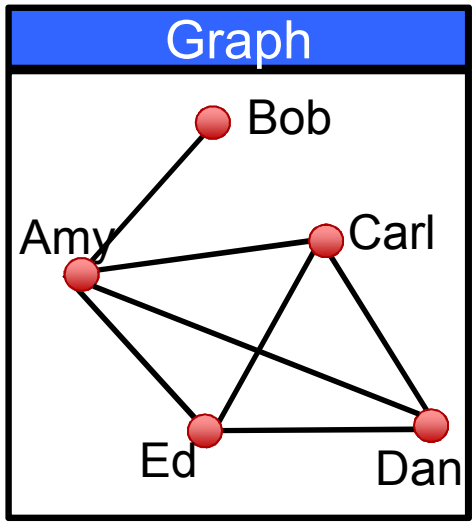
Vertices: Users

	1	2	3
Amy	1		1
Bob	1		
Carl		1	1
Dan		1	1
Ed		1	1

Relational data / hypergraph incidence matrix



Hyperedges connect 1 or more vertices



Edges connect 2 vertices

- Generalizations of graphs
  - Hyperedges represent multiway relationships between vertices
- Convenient representations of relational data
  - Each email (subset of users) can be represented by hyperedge
  - Relational data often stored as hypergraph incidence matrices (e.g., D4M* tables

*Kepner, Jeremy, et al. "Dynamic distributed dimensional data model (D4M) database and computation system." *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012.

Hypergraph to Graph: Clique Expansion

Sandia National Laboratories

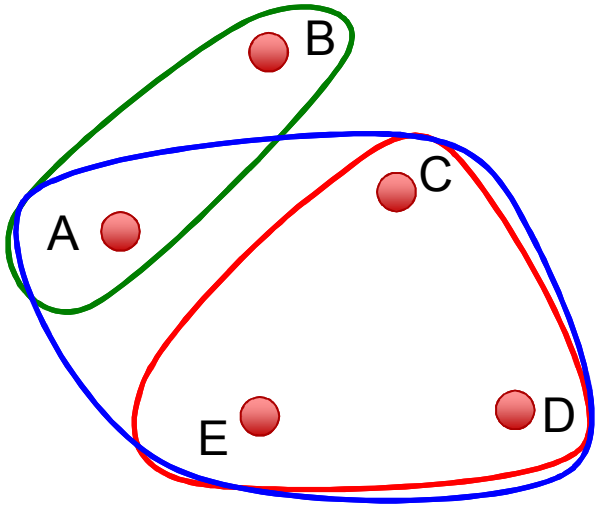
Hyperedges

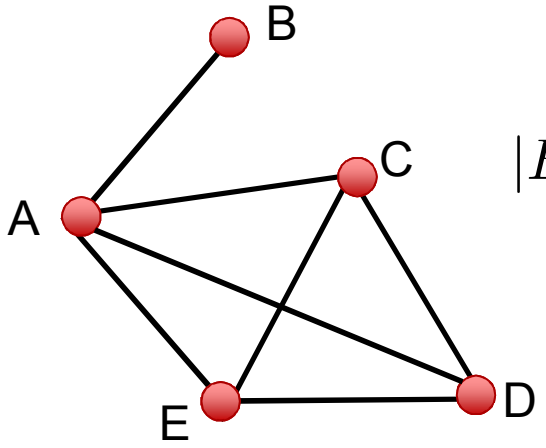
Vertices

	1	2	3
A	1		1
B	1		
C		1	1
D		1	1
E		1	1

Graph Edges

	1	2	3	4	5	6	7	8	9	10
A	1				1	1	1			
B	1									
C		1	1		1			1	1	
D		1		1		1		1		1
E			1	1			1		1	1





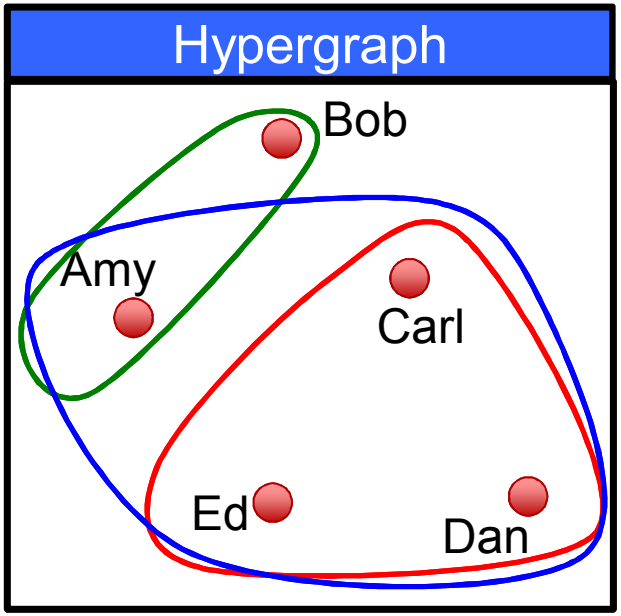
$$|E_g| = \sum_{e_h \in E_h} \binom{d(e_h)}{2}$$

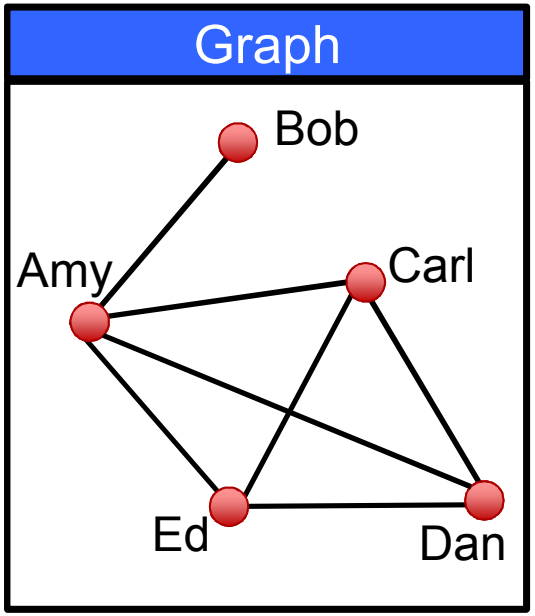
hyperedge cardinality

Graph obtained through clique expansion of hypergraph

Why hypergraphs?

	1	2	3
Amy	1		1
Bob	1		
Carl		1	1
Dan		1	1
Ed		1	1





- Typically graph models lose information
  - Were Carl, Dan, and Ed involved in same email?
  - Fix: multi-graphs + metadata, changes to algorithms

Hypergraphs represent multiway relationships unambiguously

Why hypergraphs?

1		1
1		
	1	1
	1	1
	1	1

Hypergraph incidence matrix

1				1	1	1			
1									
	1	1		1			1	1	
	1		1		1		1		1
		1	1			1		1	1

Graph Incidence matrix

$$|E_g| = \sum_{e_h \in E_h} \binom{d(e_h)}{2}$$

- Hypergraphs require significantly **less storage space** than graphs generated using clique expansion
- Hypergraph incidence matrices require **fewer operations** for matrix-vector multiplication

Hypergraphs have computational advantages

Outline

- Introduction to Hypergraphs
- Spectral clustering
- Software implementation
- Spectral clustering results
- Conclusions

Motivating Problem: Clustering of Relational Data

Sandia National Laboratories

- Determine groupings of data objects given sets of relationships amongst those objects
  - Relationships may be represented as graph or hypergraph
- Focus: **spectral clustering**
  - Compute the smallest eigenpairs of the graph or hypergraph Laplacian
  - Normalized graph Laplacian:

$$L_G = I - D_{vG}^{-1/2}(H_G H_G^T - D_{vG})D_{vG}^{-1/2}$$

  - Hypergraph Laplacian (Zhou, et al., 2006)

$$L_h = I - D_{vH}^{-1/2} H_H D_{eH}^{-1} H_H^T D_{vH}^{-1/2}$$

  - Eigenvectors used to group vertices into clusters (sorting, **kmeans++**, …)

SpMV = Sparse matrix-dense vector multiplication

Laplacians: To Form or not to Form

- Laplacians:

$$L_G = I - D_{vG}^{-1/2}(H_G H_G^T - D_{vG})D_{vG}^{-1/2}$$
$$L_h = I - D_{vH}^{-1/2} H_H D_{eH}^{-1} H_H^T D_{vH}^{-1/2}$$

- One option: Explicitly form Laplacian
- Instead: **Define application of Laplacian as series of SpMV and vector addition operations**
  - Store incidence matrix, degree matrices
  - Reason 1: Matrix-matrix products are expensive
  - Reason 2: Dynamic graphs – easier to change incidence matrix than Laplacian

Computational advantages to not explicitly forming Laplacians

SpMV = Sparse matrix-dense vector multiplication

Outline

- Introduction to Hypergraphs
- Spectral clustering
- Software implementation
- Spectral clustering results
- Conclusions

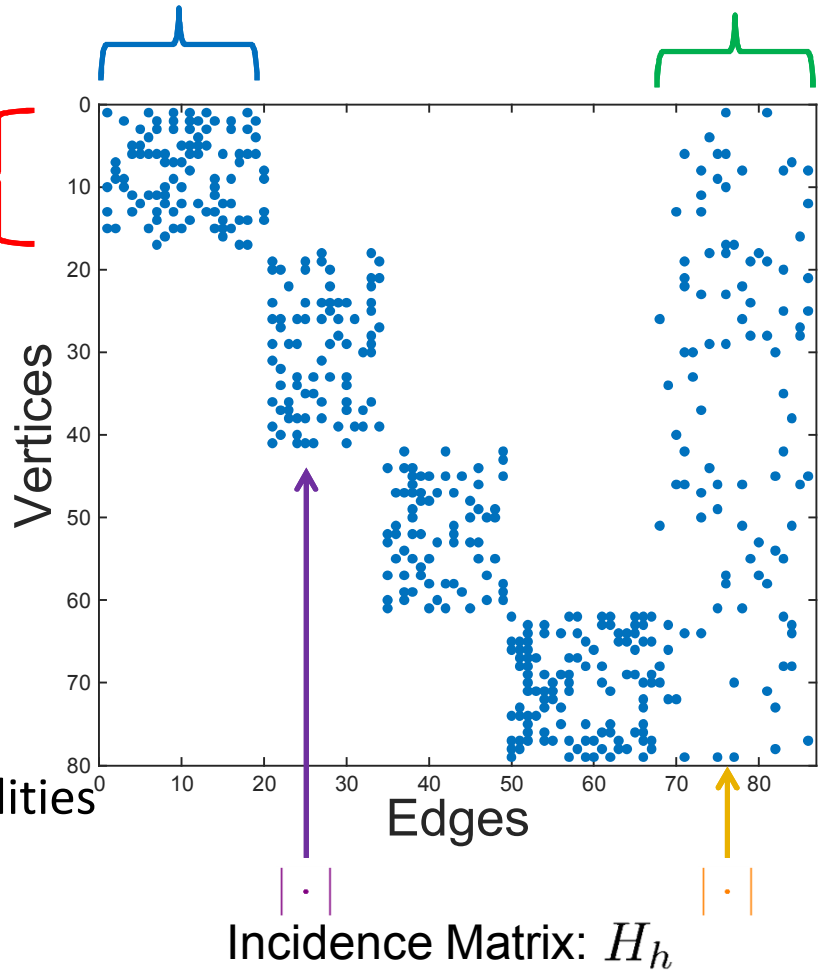
TriData: Trilinos for Data Analysis

Sandia National Laboratories

- C++ Trilinos based software for data analytics problems
  - High performance (target: billions of vertices)
  - Early focus on spectral methods
  - Hypergraphs and graphs
- Supports several eigensolvers available in Anasazi
  - LOBPCG, TraceMin-Davidson, Riemannian Trust Region, Block Krylov-Schur
  - Very fast convergence for Laplacians (especially hypergraphs)
- Trilinos/Anasazi supports modern HPC architectures
  - MPI+X
  - Where X = {multicore, GPUs, Xeon Phi…}

Hypergraph Generator

- Randomly generated hypergraphs
  - Stochastic block-like
  - Can generate multiple random instances for each parameter set
- Parameterized Generator
  - Clusters
  - Vertices per cluster
  - # of intra/inter-cluster hyperedges
  - Intra/Inter-cluster hyperedge cardinalities



- **Generate real-world inspired hypergraphs at different scales**
- **Generates "ground truth"**

Outline

- Introduction to Hypergraphs
- Spectral clustering
- Software implementation
- Spectral clustering results
  - Clustering Quality
  - Runtimes
- Conclusions

Numerical Experiments

- Experiments conducted on workstation with 128 GB of memory using 16 cores
- Generated Hypergraphs
  - Number of clusters: 5
  - Vertices per cluster (mean): 10,000
  - Intra/inter-cluster hyperedges (mean): 20,000 / 200,000
  - Intra/inter-cluster hyperedge cardinality (mean): **3-10 / 3-10**
- "Quality" of our clustering measured using the Jaccard index
  - T = true cluster assignments – "ground truth" from generator
  - P = predicted cluster assignments
  - J(T,P)=1 means matches "ground truth" exactly

$$J(T,P) = \frac{|T \cap P|}{|T \cup P|}$$

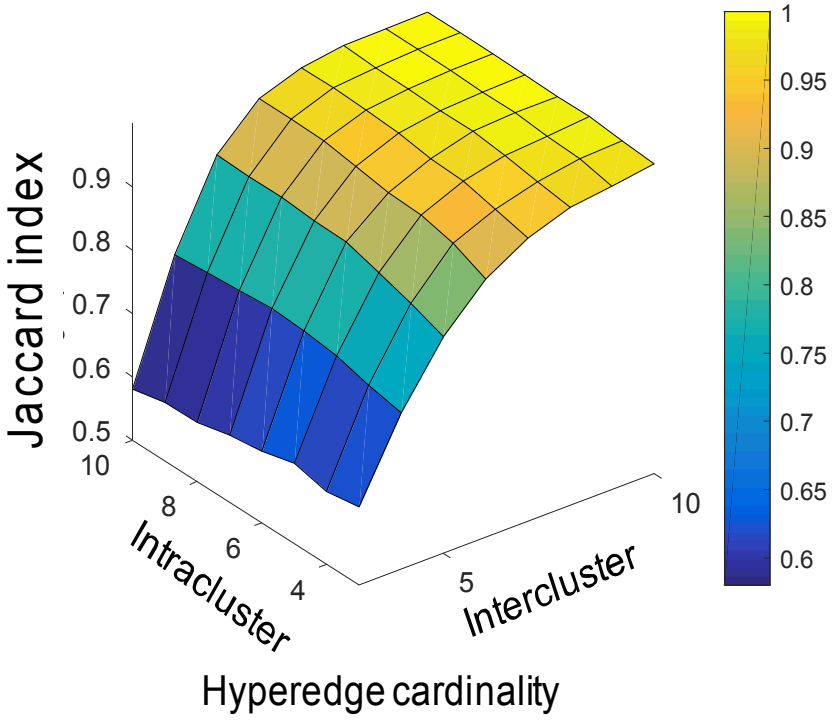
Outline

- Introduction to Hypergraphs
- Spectral clustering
- Software implementation
- Spectral clustering results
  - Clustering Quality
  - Runtimes
- Conclusions

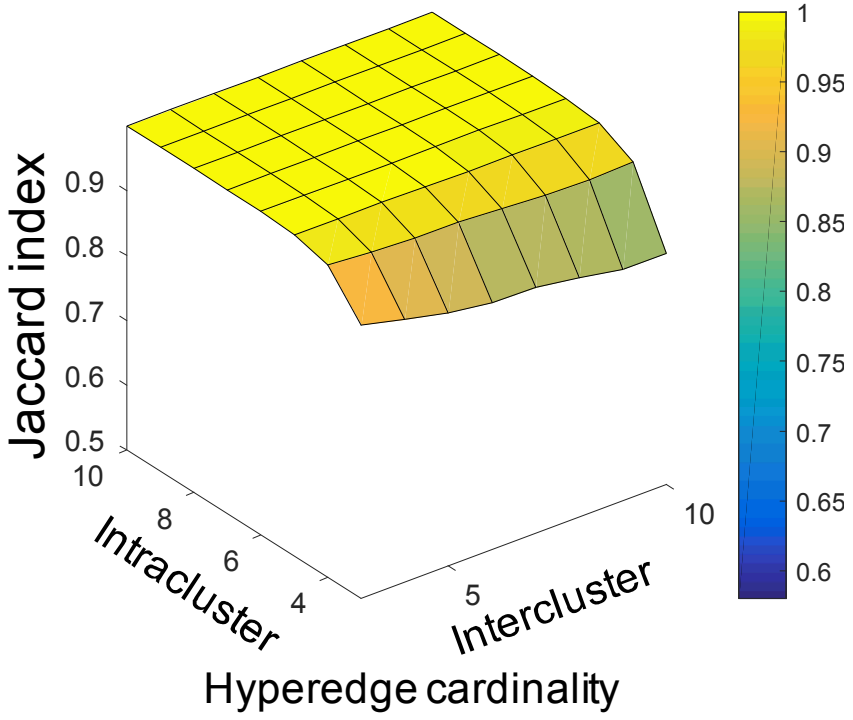
Clustering Quality: Hyperedge Cardinality

Sandia National Laboratories

Graph Model



Hypergraph Model



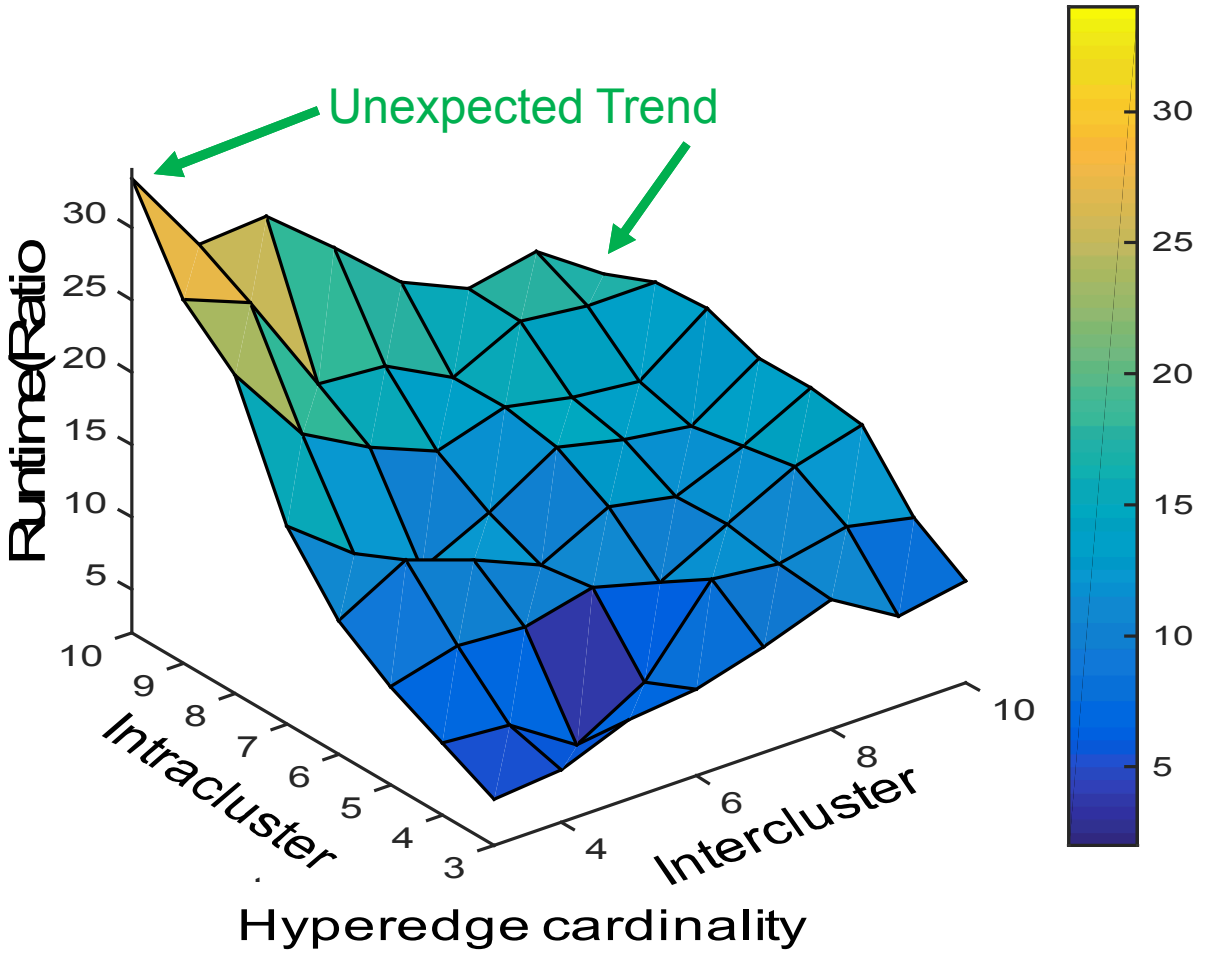
Jaccard Index=1 means clusters match ”ground truth” exactly

General trend: hypergraph based clusters more similar to “ground truth” clusters than graph based clusters

Outline

- Introduction to Hypergraphs
- Spectral clustering
- Software implementation
- Spectral clustering results
  - Clustering Quality
  - Runtimes
- Conclusions

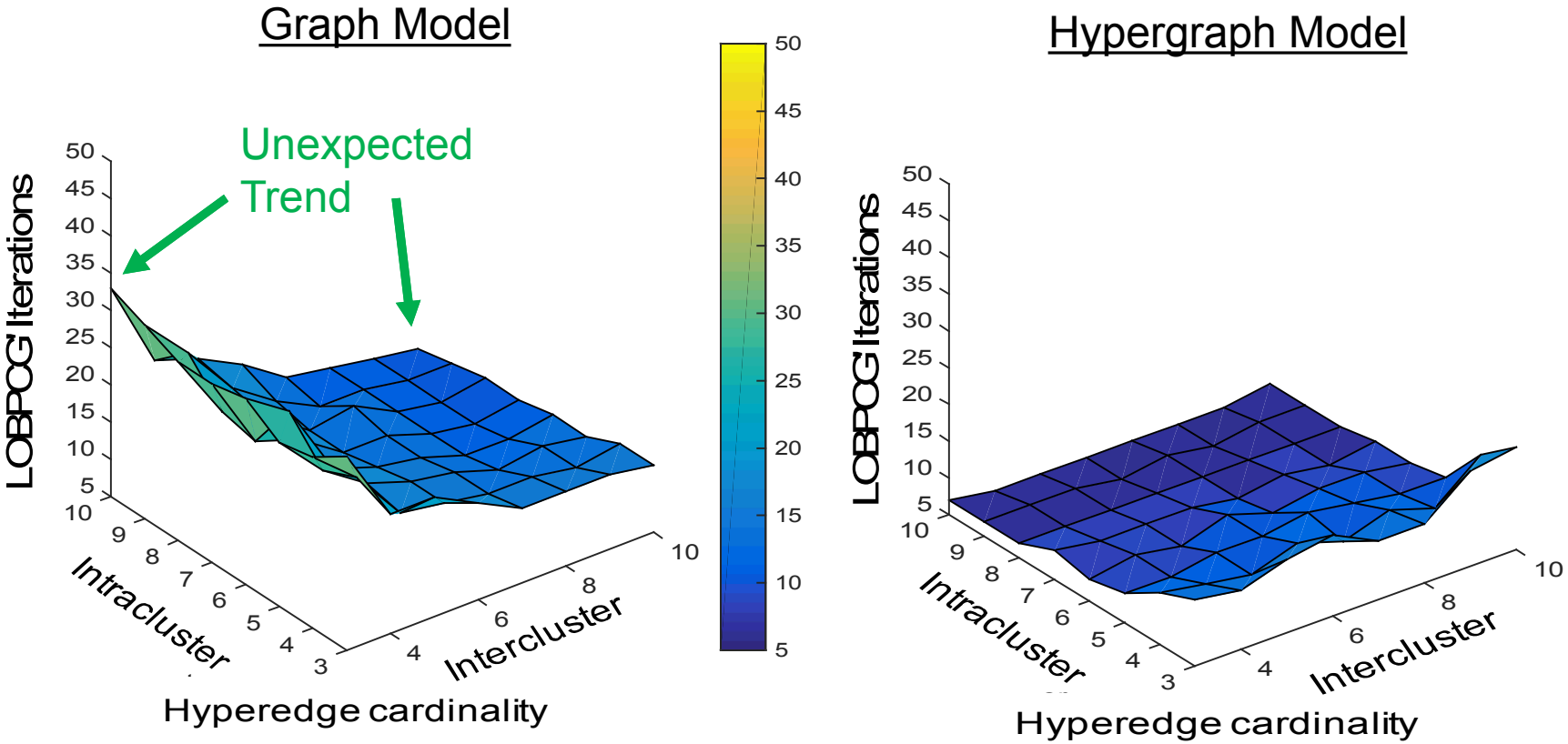
Runtimes: Hyperedge Cardinality



Runtime ratio = graph runtime / hypergraph runtime

Hypergraph models up to 30x faster than graph models

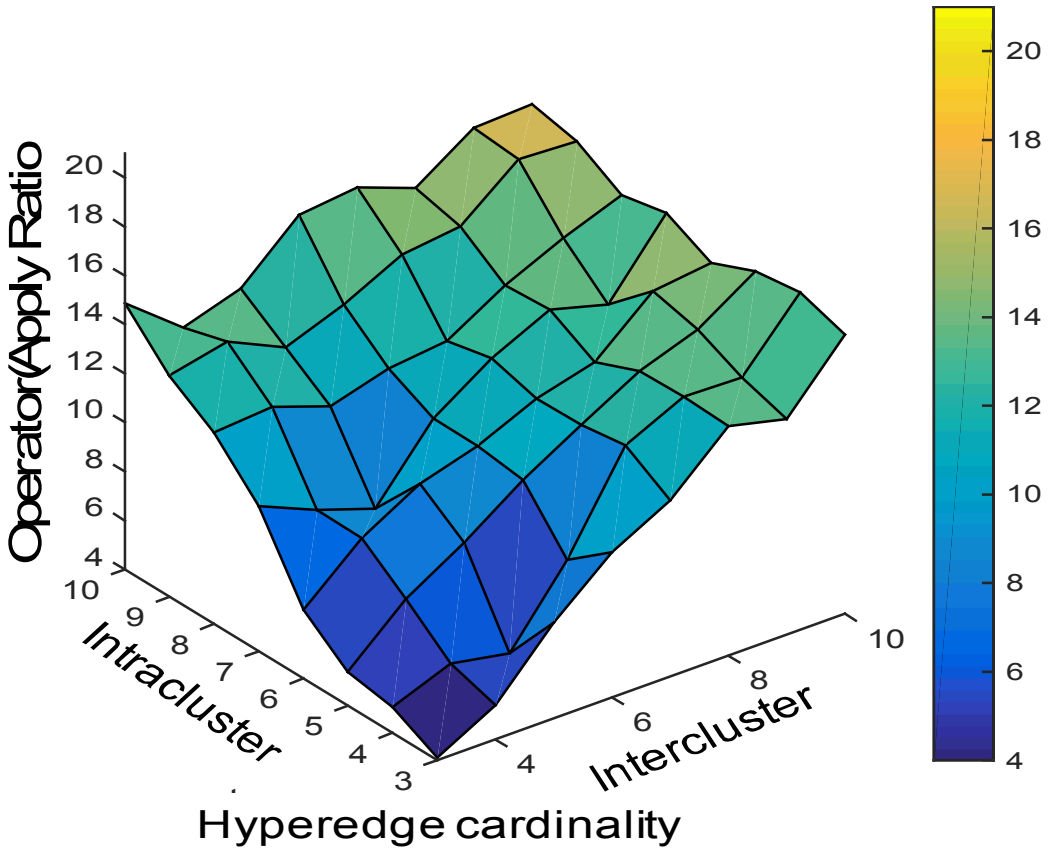
Eigensolver Iterations: Hyperedge Cardinality



- Hypergraph required fewer LOBPCG iterations than graph
  - Better separation of eigenvalues in hypergraph Laplacian

Hypergraph models converged faster (up to 6x) than graph models

Operator Apply Time: Hyperedge Cardinality



Laplacian operator apply more efficient for hypergraph model than graph model (up to 17x faster)

Outline

- Introduction to Hypergraphs
- Spectral clustering
- Software implementation
- Spectral clustering results
- Conclusions

Summary/Conclusions

Sandia National Laboratories

- Explored hypergraphs for modeling relational data
  - TriData software for Linear Algebra Based Data Analytics
  - Built on Trilinos framework (target: billions of vertices)
- Compelling advantages for computing on incidence matrices
  - Avoid forming Laplacians, adjacency matrices when possible
- Hypergraphs **extremely** promising for data analytics
  - Better model than graphs for multiway relational data (spectral analysis)
  - **Improvements in quality**
  - **Significant improvements in runtime** (30x)

Future Work

Sandia National Laboratories

- Target larger problems
  - Currently solving problems of O(100k) vertices
  - Eventual target: O(1 billion) vertices
- Performance improvements to software
  - 2D partitioning – important for scalability beyond O(10k) processors
  - Improved efficiency of parallel hypergraph generator
- **Promising problems for hypergraphs**: community and anomaly detection; centrality analysis; information propagation; directed hypergraphs for complex, causal relationships

Acknowledgements/References

- Thanks
  - Jon Berry, Rich Lehoucq

- Spectral Clustering
  - Ng, Andrew Y., Michael I. Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm." *Advances in neural information processing systems* 2 (2002): 849-856.
- Hypergraph Laplacian
  - Zhou, Dengyong, Jiayuan Huang, and Bernhard Schölkopf. "Learning with hypergraphs: Clustering, classification, and embedding." *Advances in neural information processing systems*. 2006.
- Related Hypergraph Work
  - Bonacich, Phillip, Annie Cody Holdren, and Michael Johnston. "Hyper-edges and multidimensional centrality." *Social networks* 26.3 (2004): 189-203.
  - Tsuda, Koji. "Propagating distributions on a hypergraph by dual information regularization." *Proc. of 22nd Int. Conf. on Machine learning*. ACM, 2005.
  - Klamt, Steffen, Utz-Uwe Haus, and Fabian Theis. "Hypergraphs and cellular networks." *PLoS Comput Biol* 5.5 (2009): e1000385.

Extra

Sandia National Laboratories

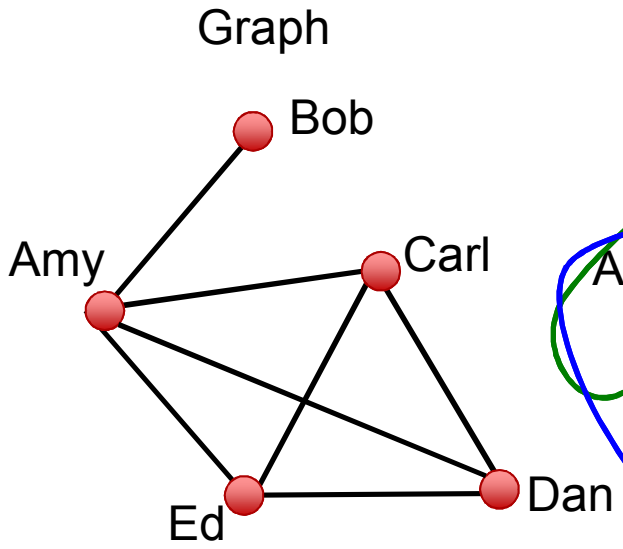
What is a hypergraph?

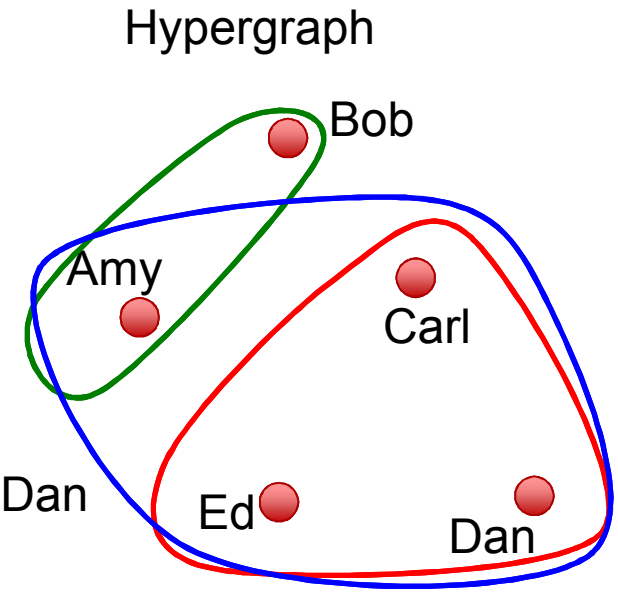
Sandia National Laboratories

Emails

Users	1	2	3
Amy	x		x
Bob	x		
Carl		x	x
Dan		x	x
Ed		x	x

Relational Data





- Convenient representation of relational data
  - Each email can be represented by hyperedge
  - Emails/hyperedges consist of subsets of users

What is a hypergraph?

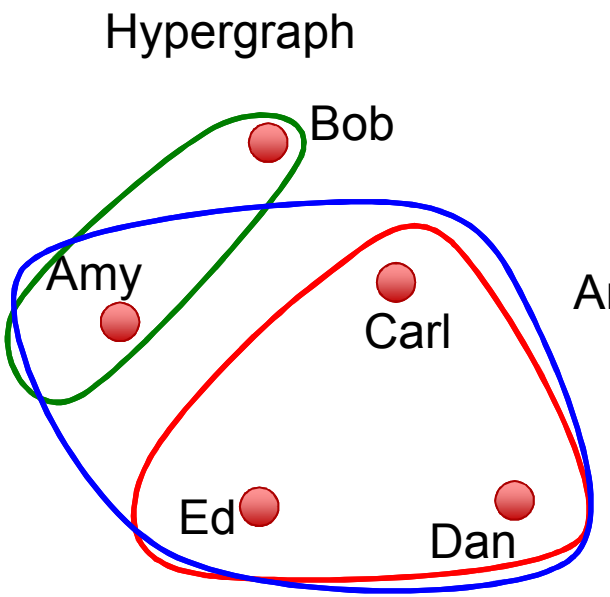
Sandia National Laboratories

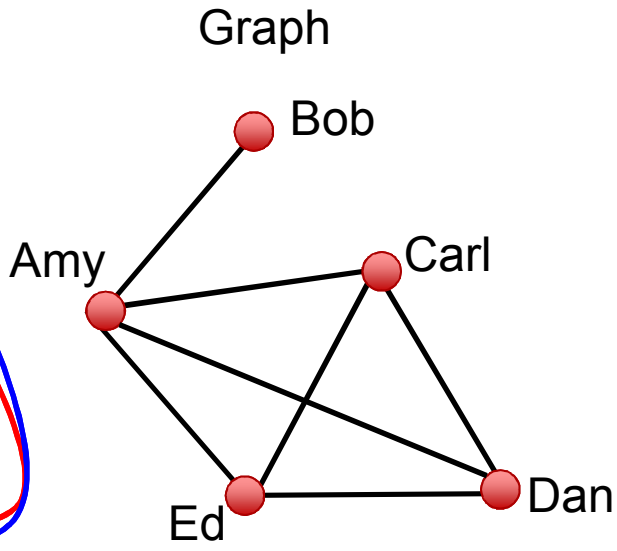
Hyperedges: Emails

Vertices: Users

	1	2	3
Amy	1		1
Bob	1		
Carl		1	1
Dan		1	1
Ed		1	1

Hypergraph incidence matrix





- Relational data is often stored as hypergraph incidence matrix
  - E.g., D4M* tables
- Graph obtained through clique expansion of hypergraph

*Kepner, Jeremy, et al. "Dynamic distributed dimensional data model (D4M) database and computation system." *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012.

Hypergraph clique expansion

Sandia National Laboratories

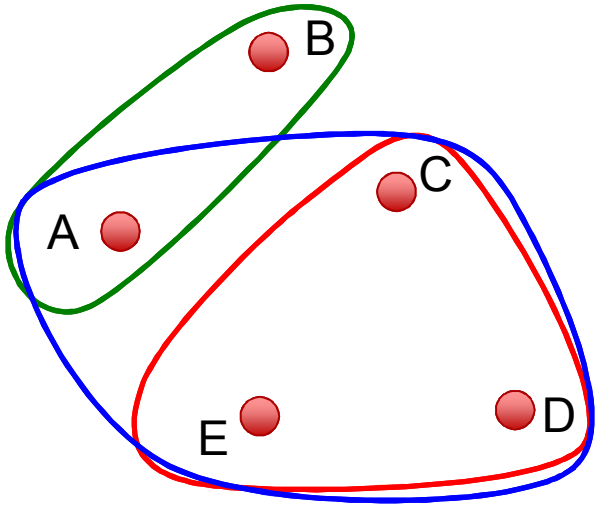
Hyperedges

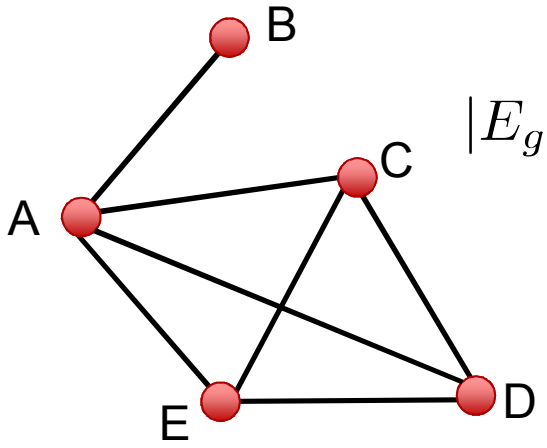
Vertices

	1	2	3
A	1		1
B	1		
C		1	1
D		1	1
E		1	1

Graph Edges

	1	2	3	4	5	6	7	8	9	10
A	1				1	1	1			
B	1									
C		1	1		1			1	1	
D		1		1		1		1		1
E			1	1			1		1	1





$$|E_g| = \sum_{e_h \in E_h} \binom{d(e_h)}{2}$$

hyperedge cardinality

Weighted hypergraph clique expansion

Sandia National Laboratories

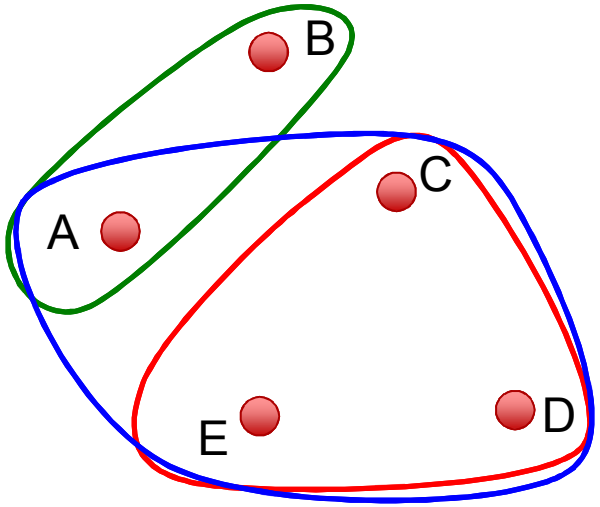
Hyperedges

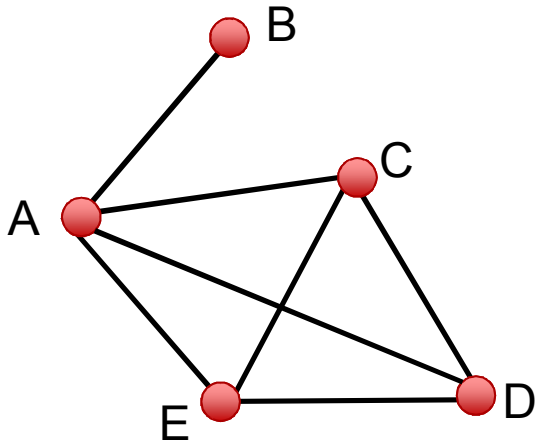
Vertices

	1	2	3
A	1		1
B	1		
C		1	1
D		1	1
E		1	1

Graph Edges

	1	2	3	4	5	6	7	8	9	10
A	1				⅓	⅓	⅓			
B	1									
C		½	½		⅓			⅓	⅓	
D		½		½		⅓		⅓		⅓
E			½	½			⅓		⅓	⅓





$$w(e_g) = \frac{1}{d(e_h) - 1}$$

hyperedge cardinality

Motivating Problem: Clustering Relational Data

Sandia National Laboratories

- Determine groupings of data objects given sets of relationships amongst those objects
- Relationships may be represented as graph or hypergraph
  - Graphs represent pairwise relationships
  - Hypergraphs represent relationships among groups of things
- Applications
  - Finding emerging research trends from documents (Jung et al., 2014)
  - Clustering categorical data (Gibson et al., 2000)
  - Image segmentation (Agarwal et al., 2005)
  - Metabolic networks (Guimera et al., 2004)

Related Hypergraph Work

Sandia National Laboratories

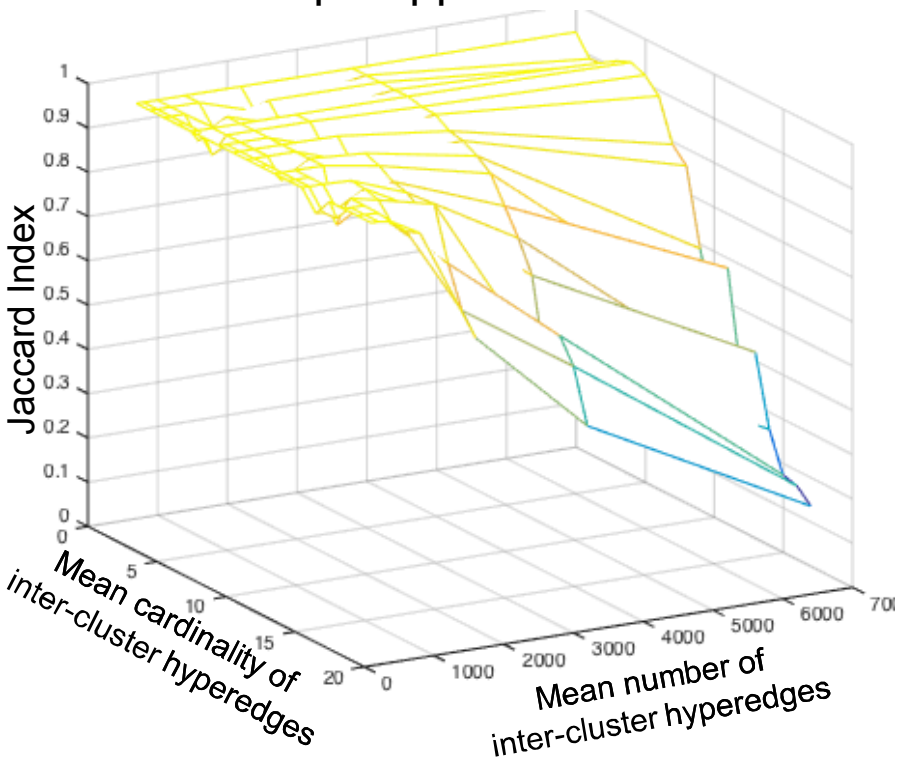
- Data partitioning (Computational science & engineering)
  - Circuit layout, parallel processing, …
- Hypergraph eigencentrality (Bonacich, et al., 2004)
  - Analysis of attacks on Caribbean settlements
- Semi-supervised Learning (Tsuda, 2005)
  - Information propagation across hypergraph, prior knowledge
- Biological networks (Klamt, et al., 2009)
  - Biochemical reactions, finding related proteins (functional groups)

- Caveat emptor – hypergraphs
  - Representations may increase complexity of algorithms (e.g., MST)
  - Avoid such problems

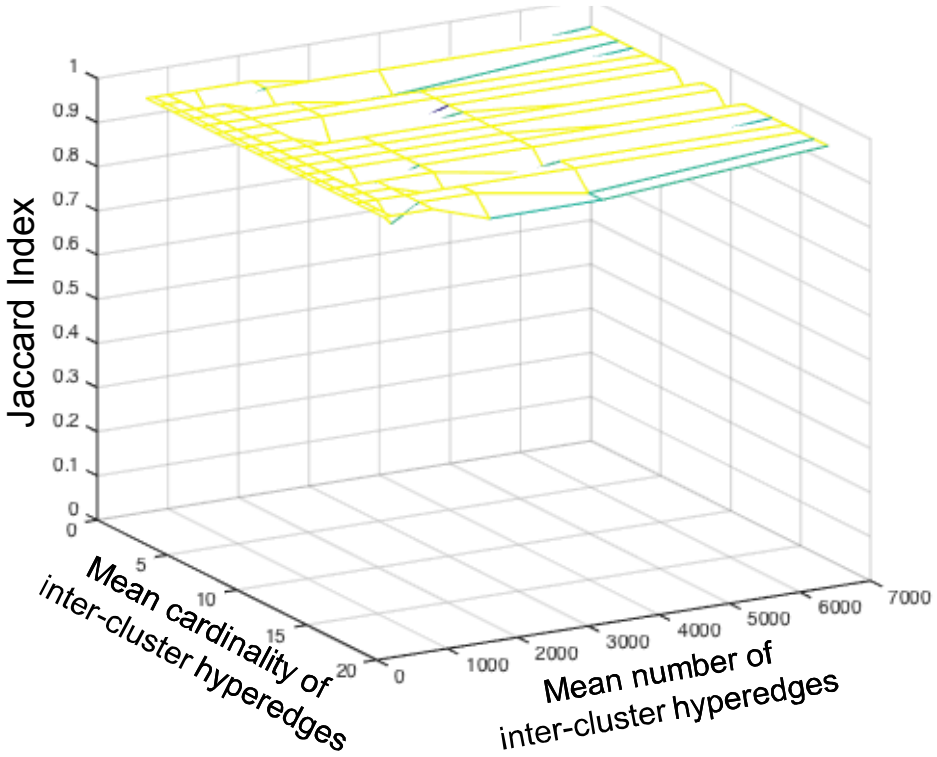
Unweighted graph vs hypergraph clustering

Sandia National Laboratories

Graph Approach



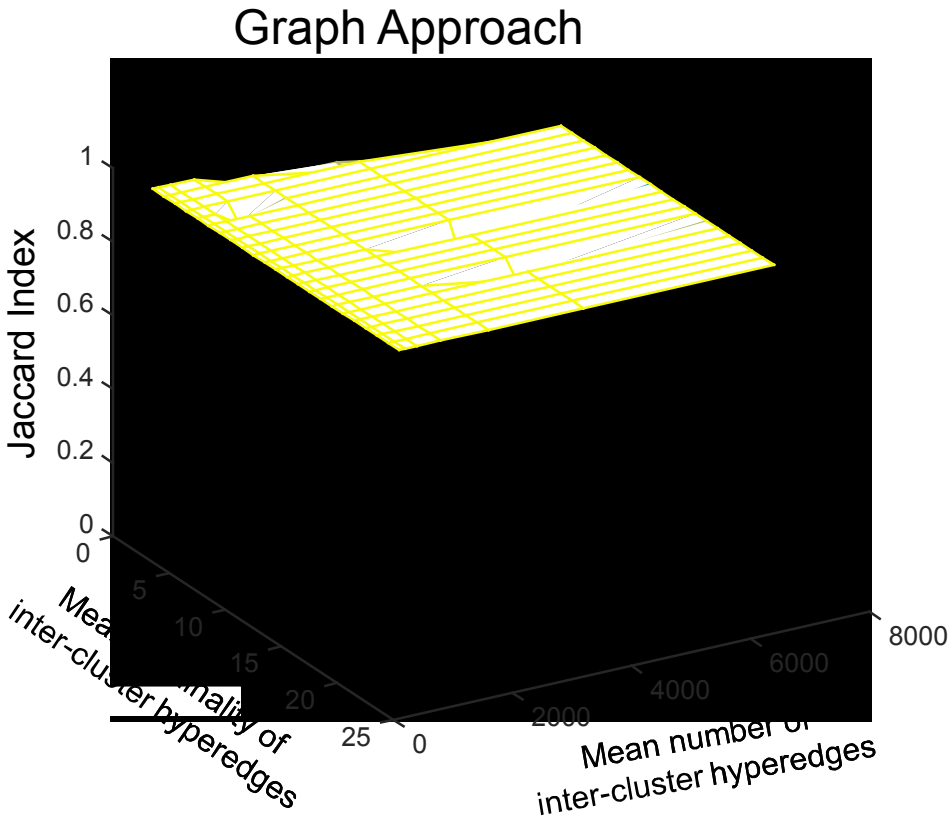
Hypergraph Approach

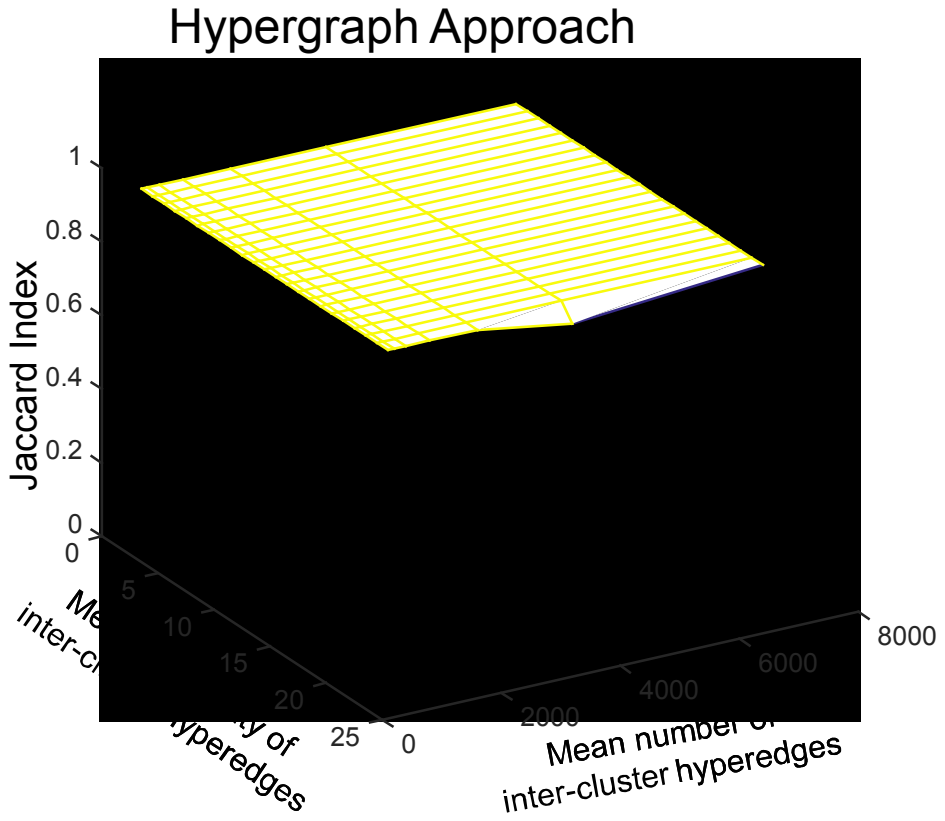


Mean intra-cluster cardinality = 3
Number of clusters 4, mean size 100
Mean number of inter-cluster edges: 800

Weighted graph vs hypergraph clustering

Sandia National Laboratories



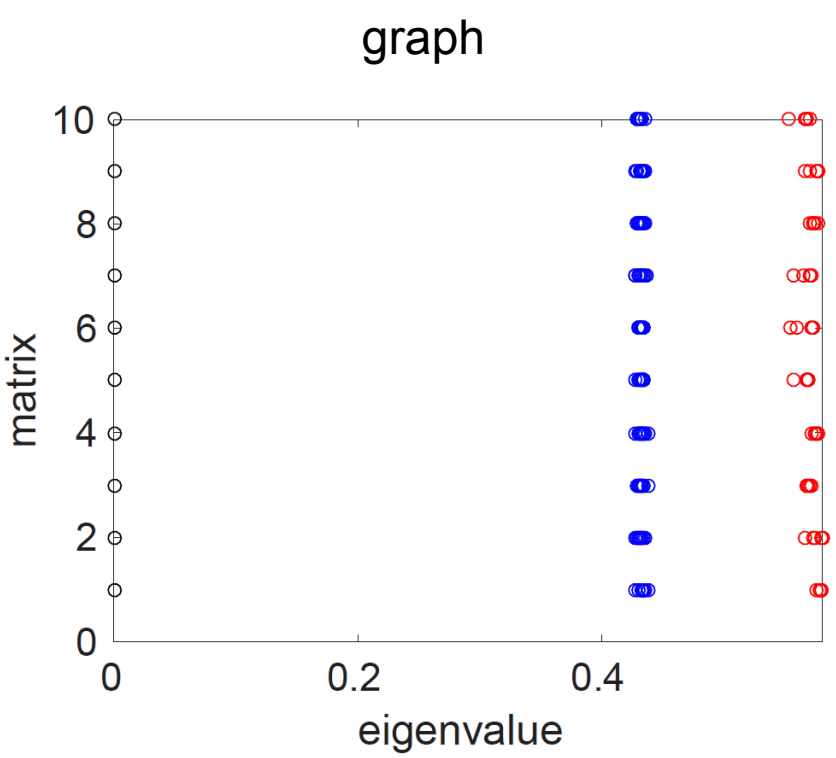


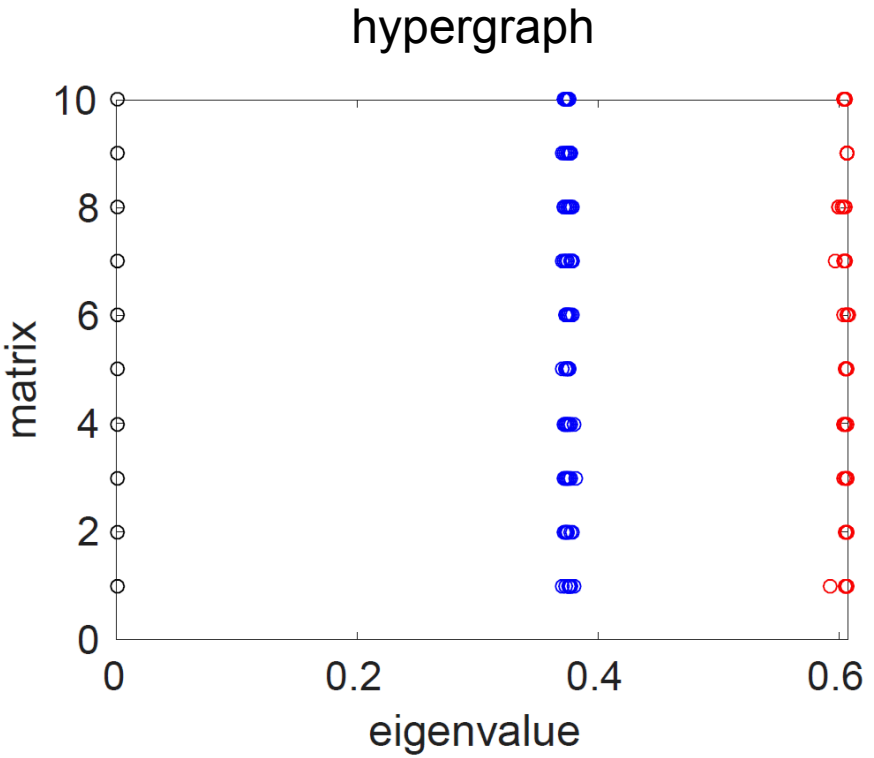
Number of clusters = 4
Mean number of vertices = 100
Number of eigenvectors = 5

Plot of eigenvalues

Sandia National Laboratories

P3





Number of clusters	10
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	20,000 200,000
Intra/Inter-cluster h-edge cardinality	5 5

Formulation of the eigenvalue problem

Sandia National Laboratories

- Computing the smallest eigenpairs of

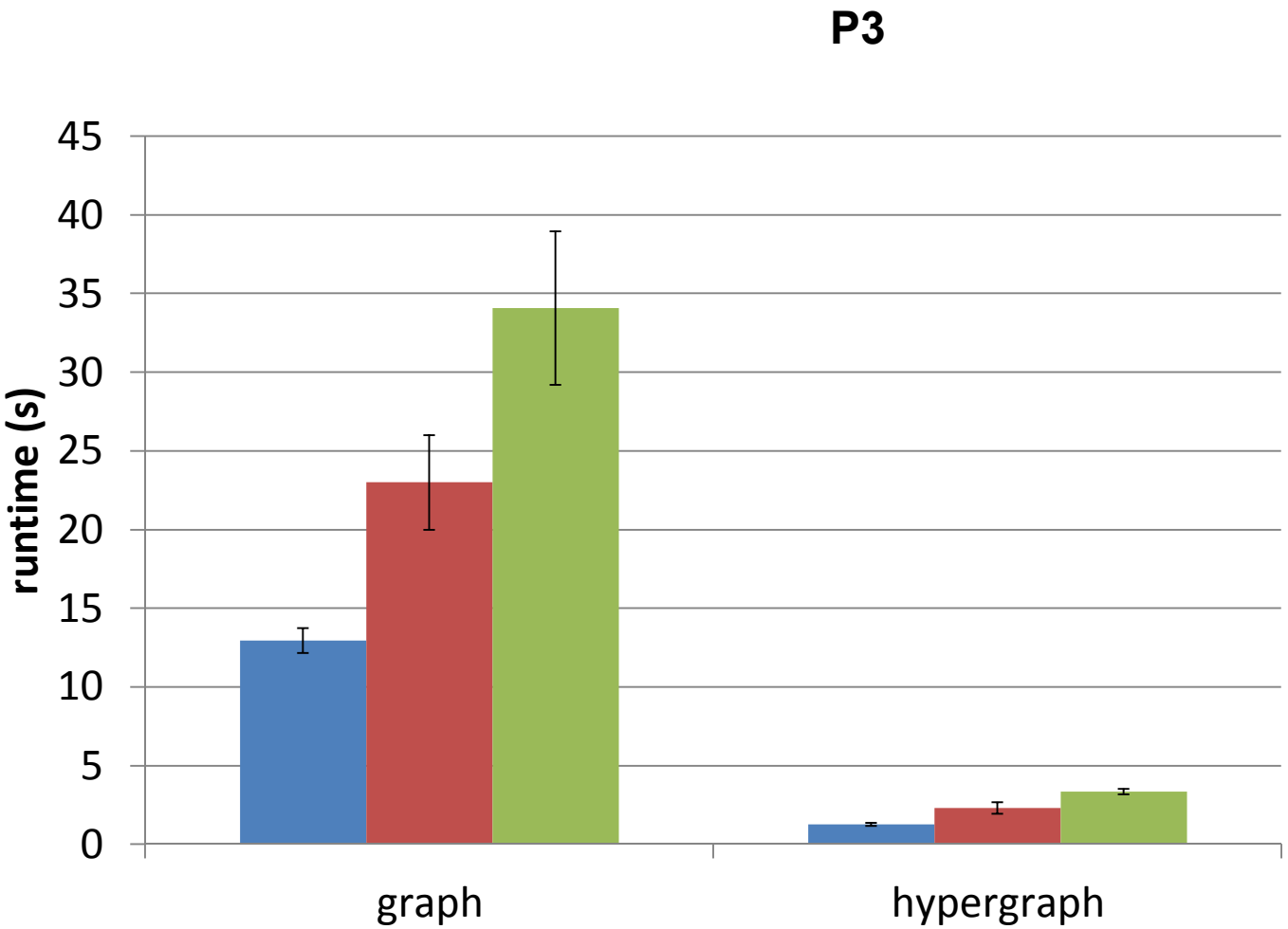
$$L_G = I - D_{vG}^{-1/2}(H_G H_G^T - D_{vG})D_{vG}^{-1/2} \qquad L_h = I - D_{vH}^{-1/2} H_H D_{eH}^{-1} H_H^T D_{vH}^{-1/2}$$

  is equivalent to computing the largest eigenpairs of the shifted Laplacians

$$S_G = D_{vG}^{-1/2}(H_G H_G^T - D_{vG})D_{vG}^{-1/2} \qquad S_h = D_{vH}^{-1/2} H_H D_{eH}^{-1} H_H^T D_{vH}^{-1/2}$$

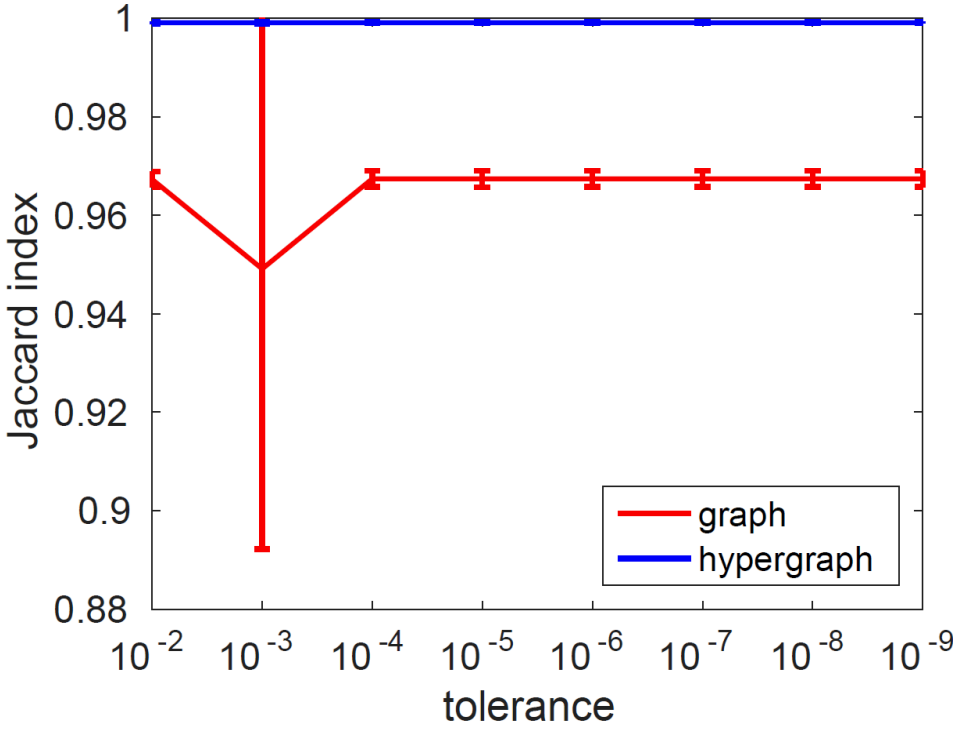
  - Computing the largest eigenpairs tends to be cheaper
  - Laplacians are singular (but null space is known)
  - L_g, L_h, and S_h are symmetric positive definite, but S_g is not
  - S_g can be shifted even more to make it positive definite

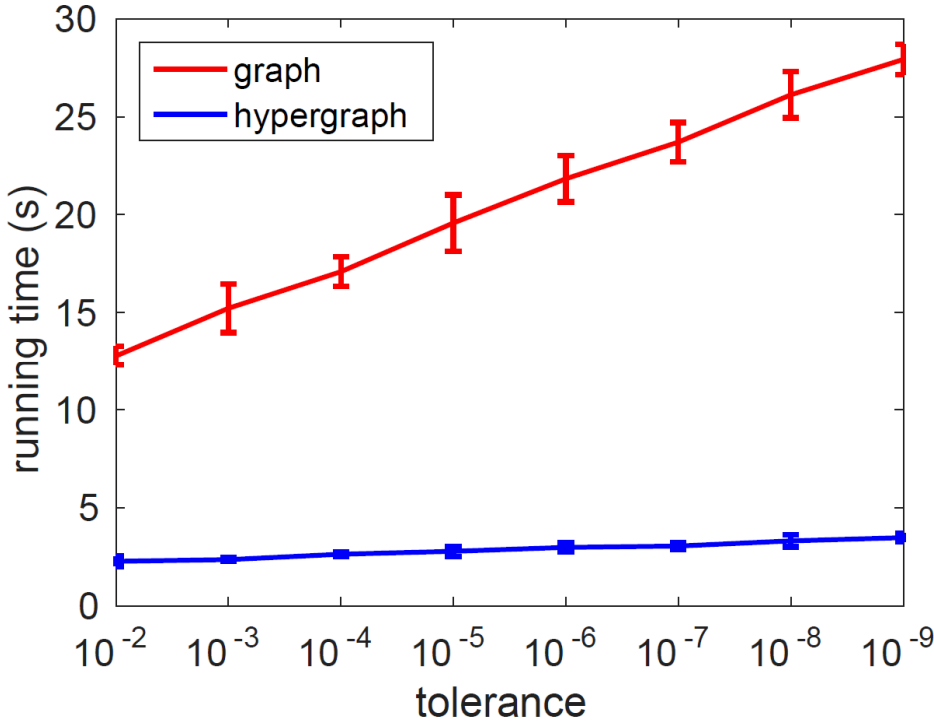
Comparison of eigensolvers



Number of clusters	10
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	20,000 200,000
Intra/Inter-cluster h-edge cardinality	5 5

What tolerance should we use?

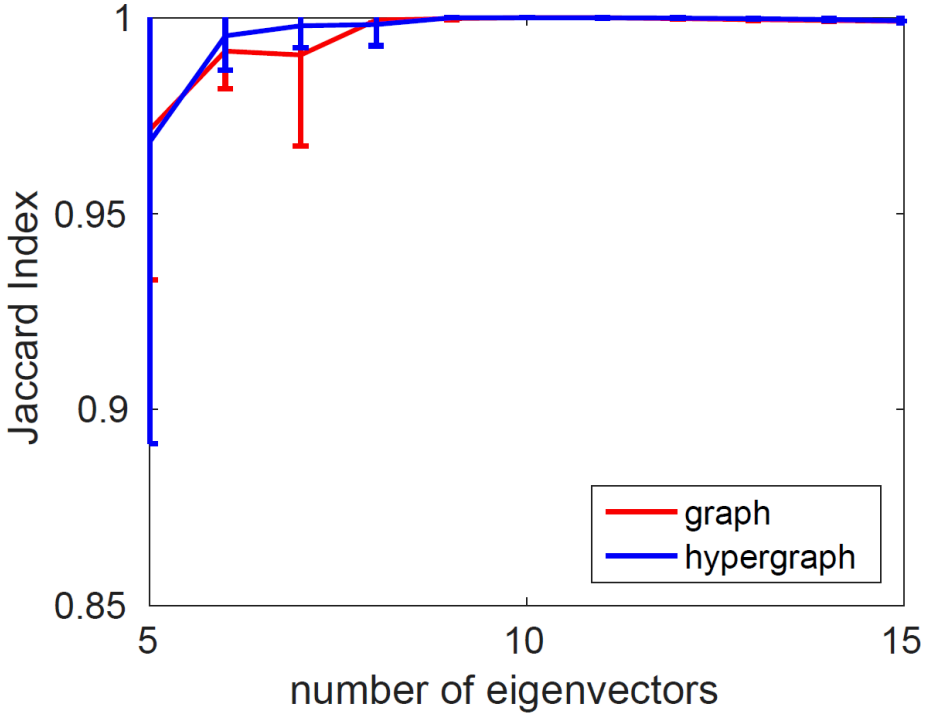


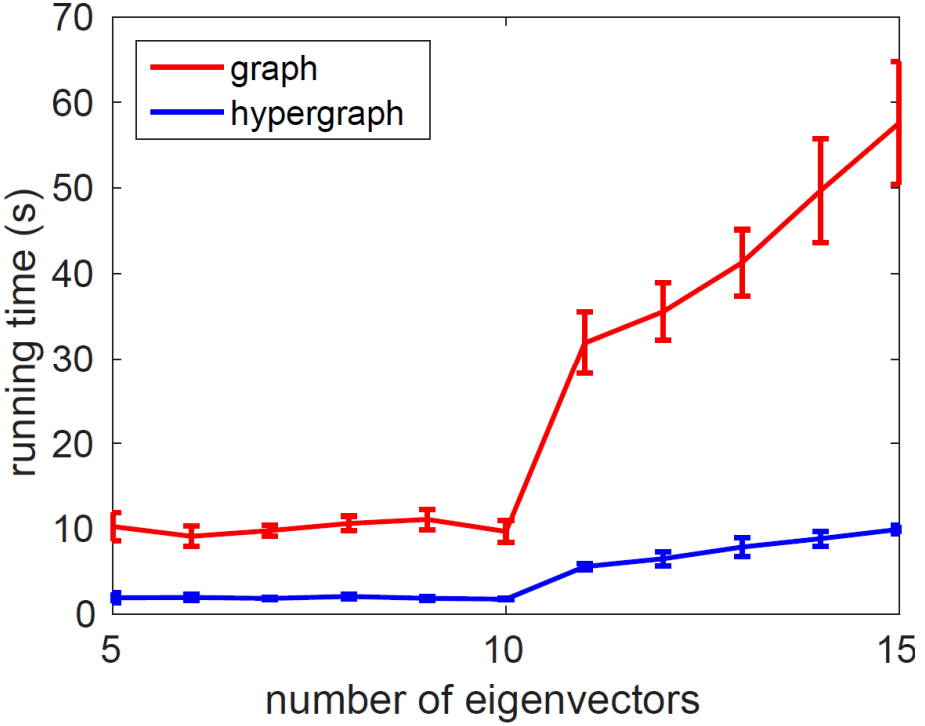


P3

Number of clusters	10
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	20,000 200,000
Intra/Inter-cluster h-edge cardinality	5 5

How many eigenvectors should we calculate?

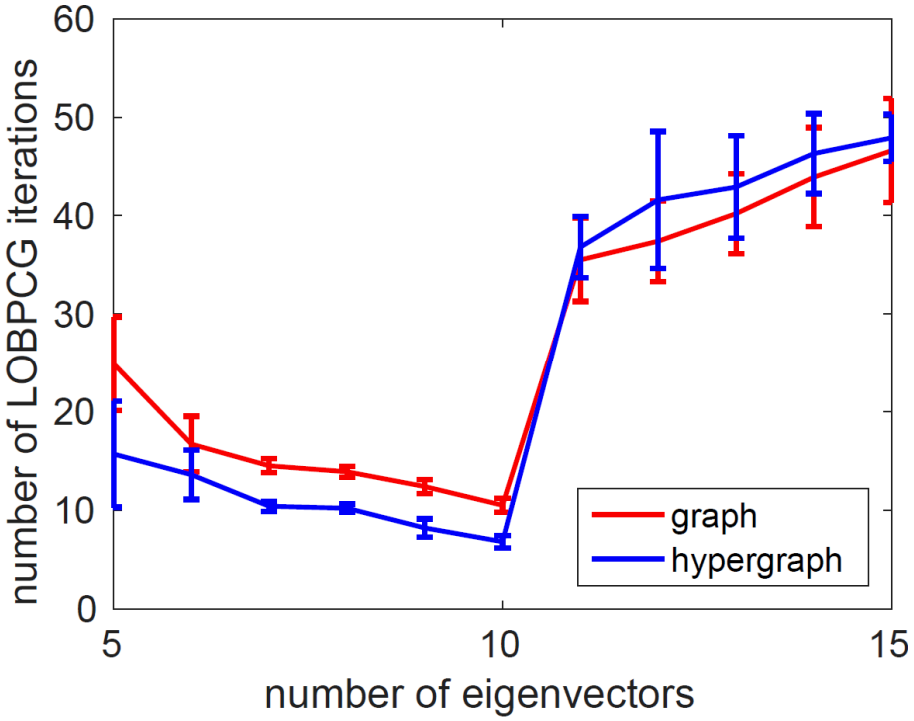




Less noisy data: P1

Number of clusters	10
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	40,000 50,000
Intra/Inter-cluster h-edge cardinality	5 5

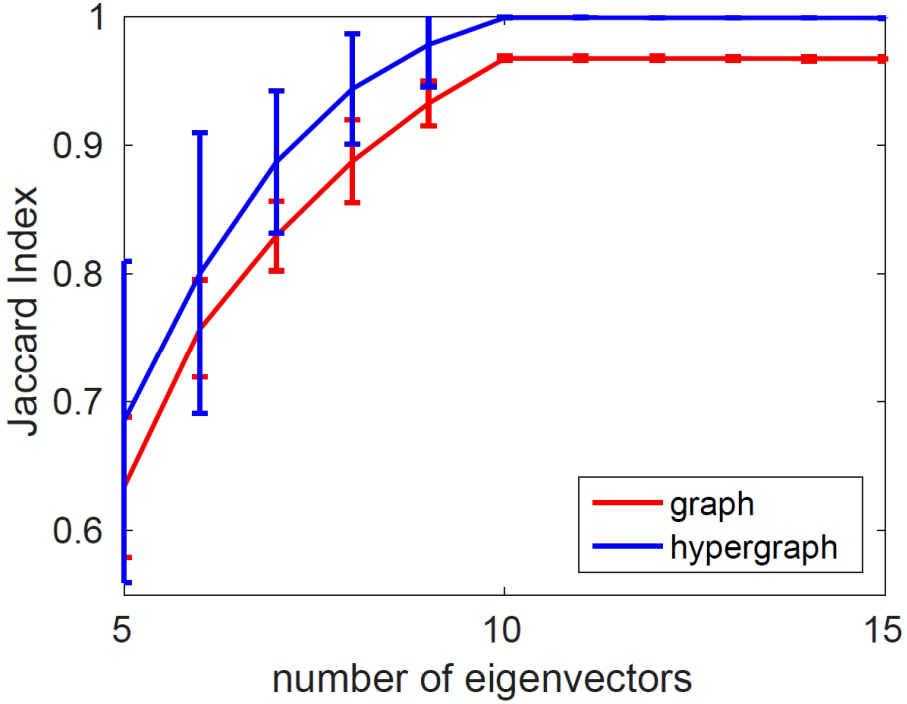
How many eigenvectors should we calculate?

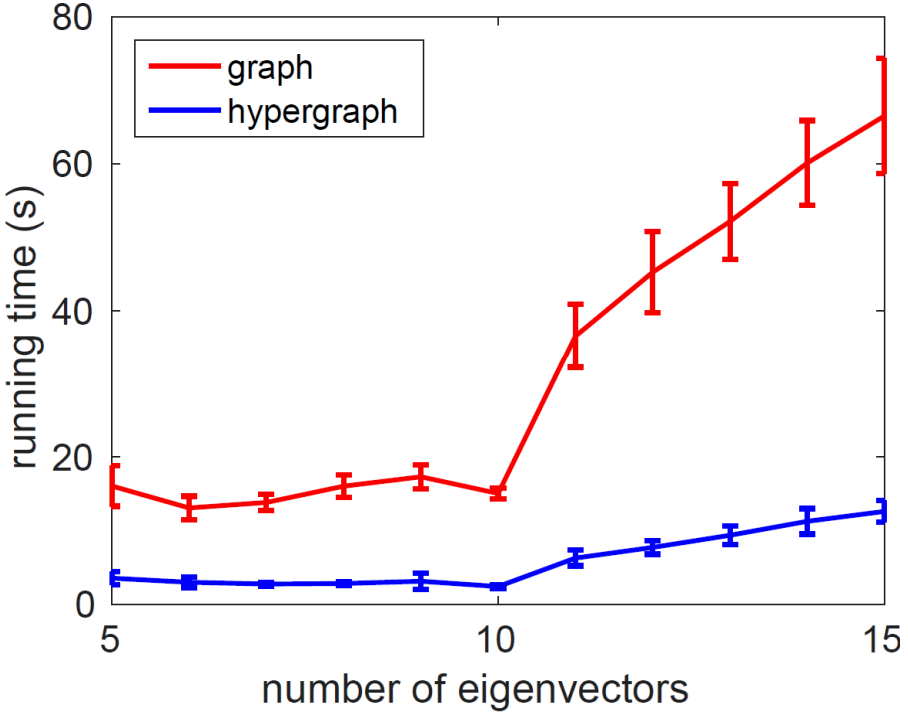


Less noisy data: P1

Number of clusters	10
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	40,000 50,000
Intra/Inter-cluster h-edge cardinality	5 5

How many eigenvectors should we calculate?

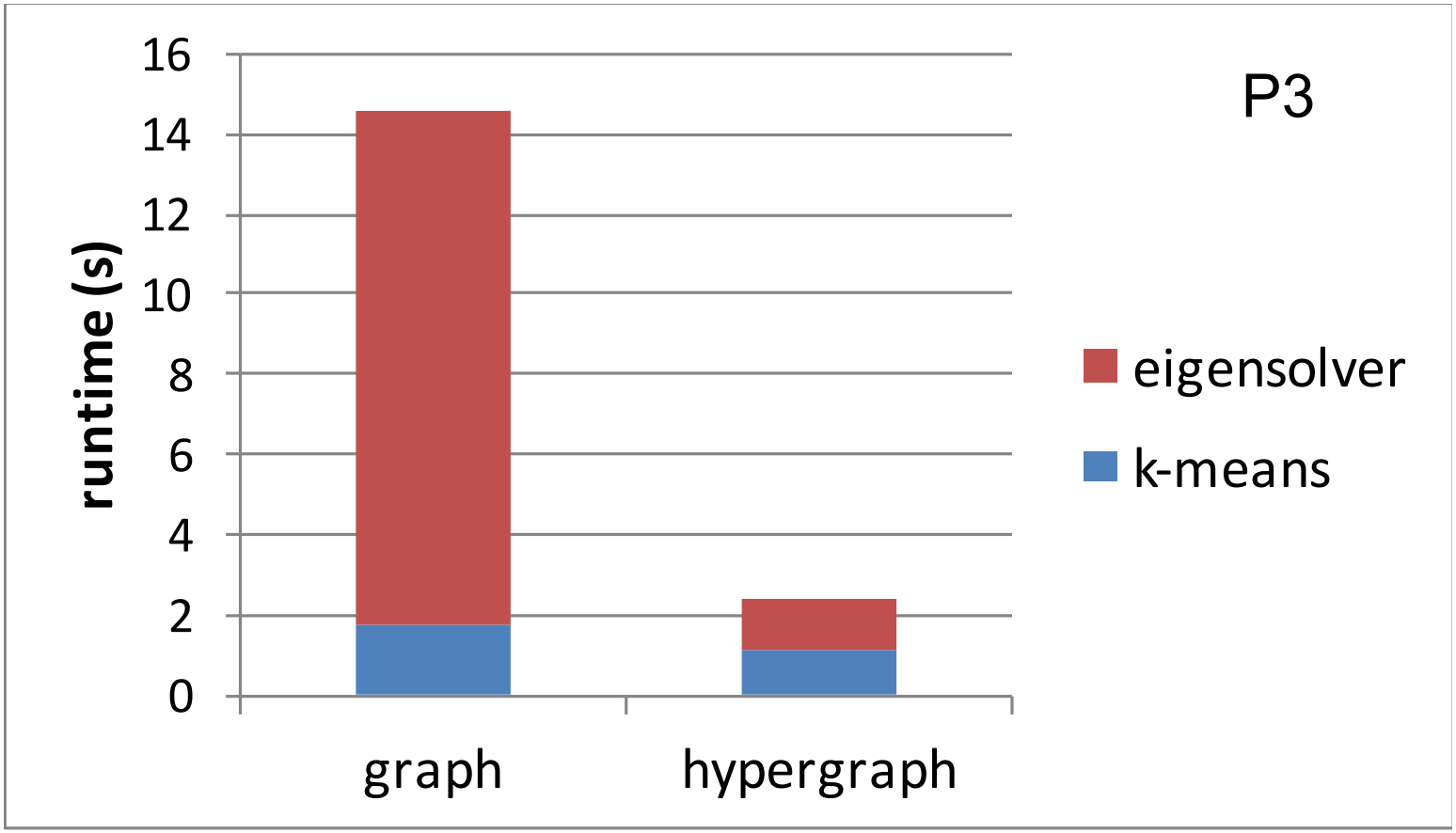




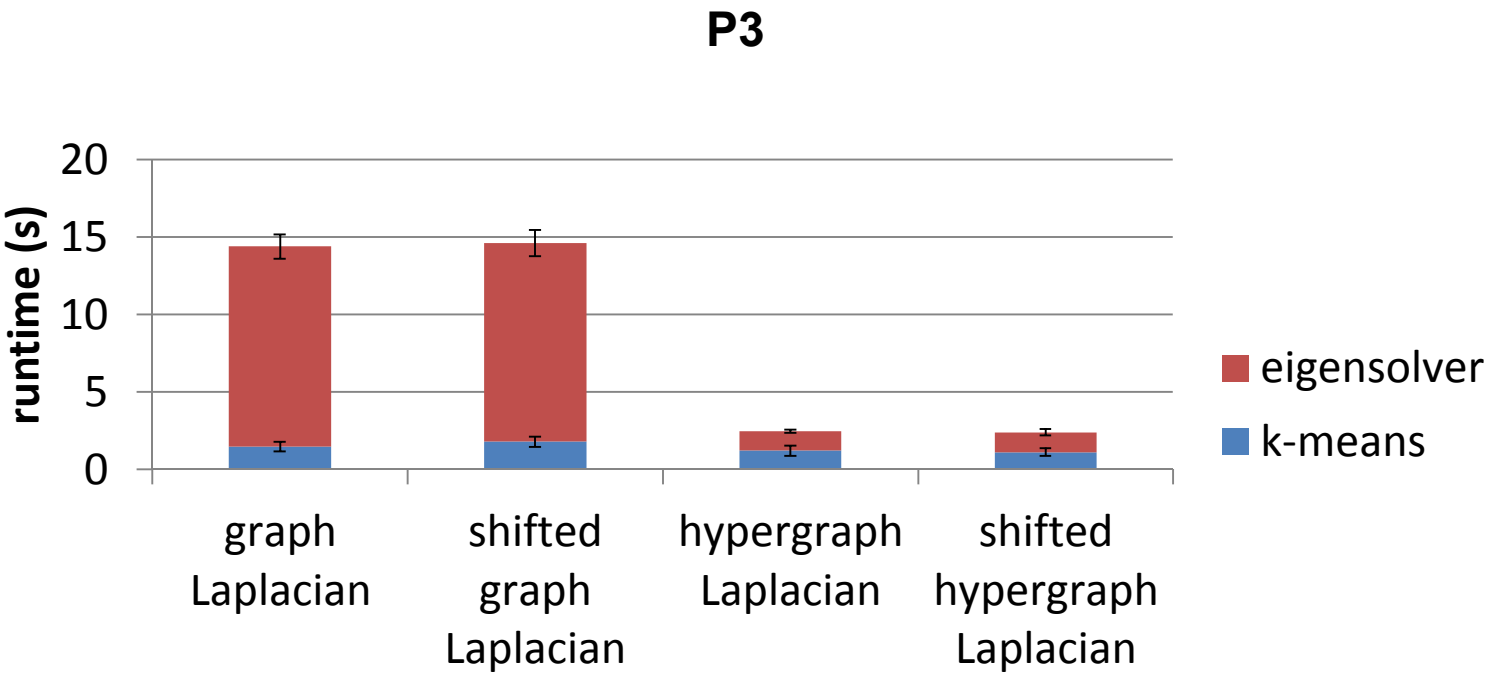
Noisier data: P3

Number of clusters	10
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	20,000 200,000
Intra/Inter-cluster h-edge cardinality	5 5

Runtimes: Eigensolver vs. K-means



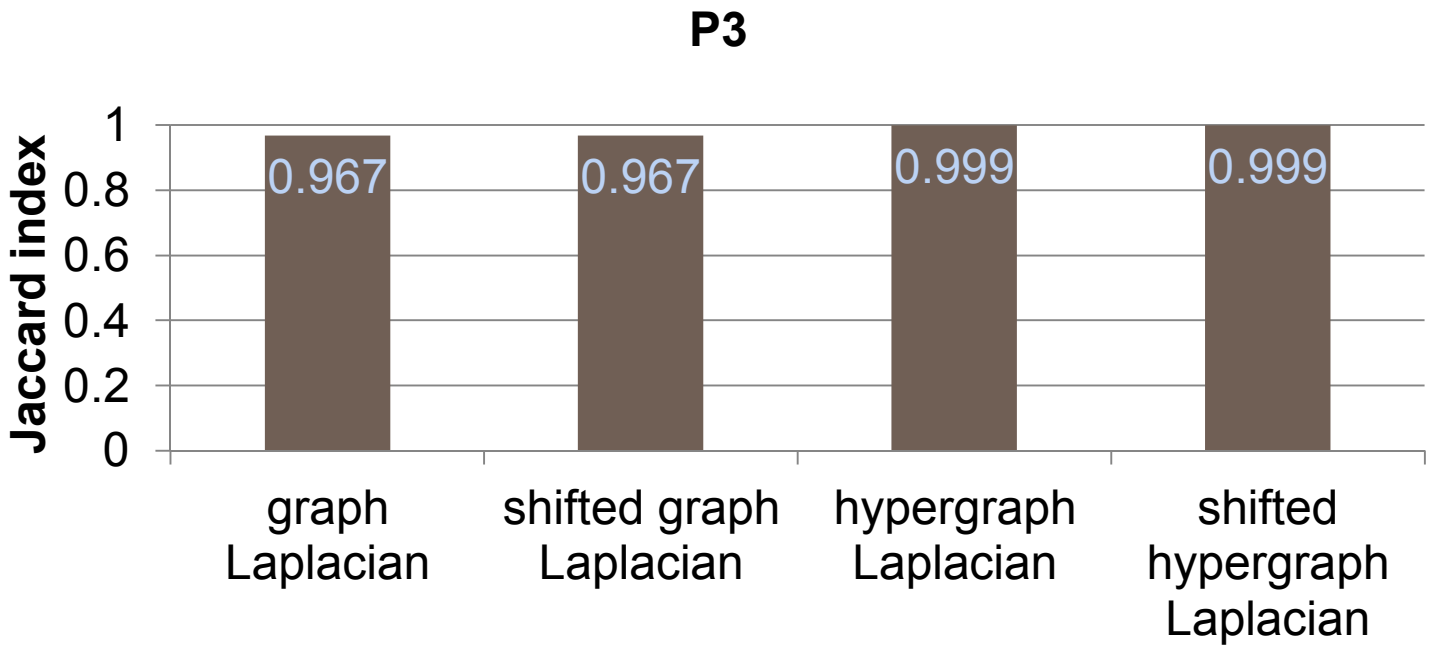
Should we compute the eigenpairs of the Laplacian or the shifted Laplacian?



	Graph Laplacian L_g	Shifted graph Laplacian S_g	Hypergraph Laplacian L_h	Shifted hypergraph Laplacian S_h
LOBPCG iterations	15.6	15.6	8.9	8.9
K-means iterations	56.9	79.4	31.8	28.1

Number of clusters	10
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	20,000 200,000
Intra/Inter-cluster h-edge cardinality	5 5

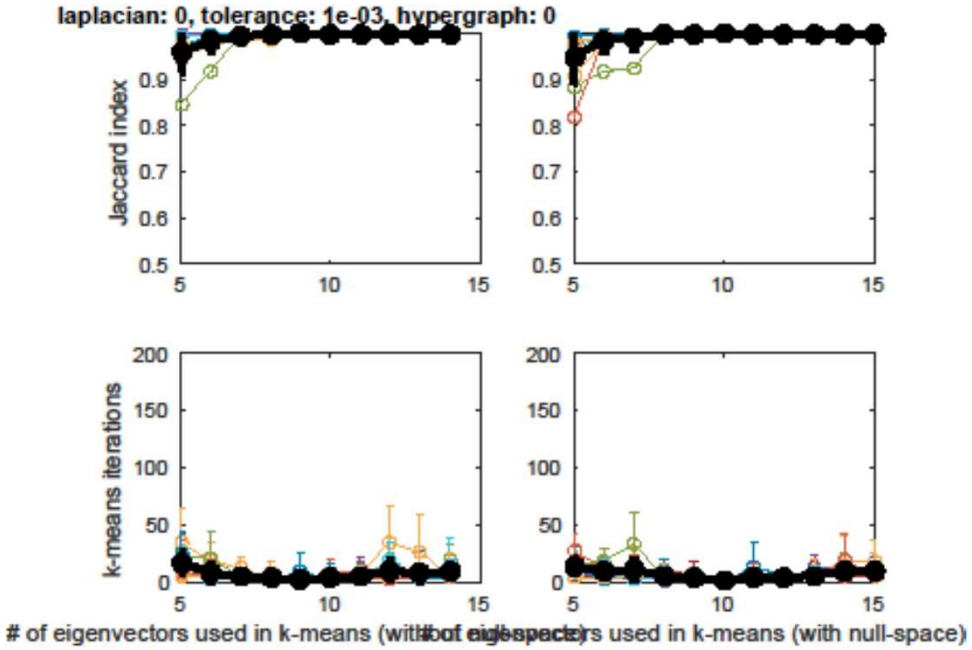
Should we compute the eigenpairs of the Laplacian or the shifted Laplacian?



	Graph Laplacian L_g	Shifted graph Laplacian S_g	Hypergraph Laplacian L_h	Shifted hypergraph Laplacian S_h
LOBPCG iterations	15.6	15.6	8.9	8.9
K-means iterations	56.9	79.4	31.8	28.1

Number of clusters	10
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	20,000 200,000
Intra/Inter-cluster h-edge cardinality	5 5

Should the null space be provided to k-means?



- nClusters: 10
- nVerts: 10,000
- nEdges in cluster: 40,000
- nEdges between clusters: 50,000

Data Sets

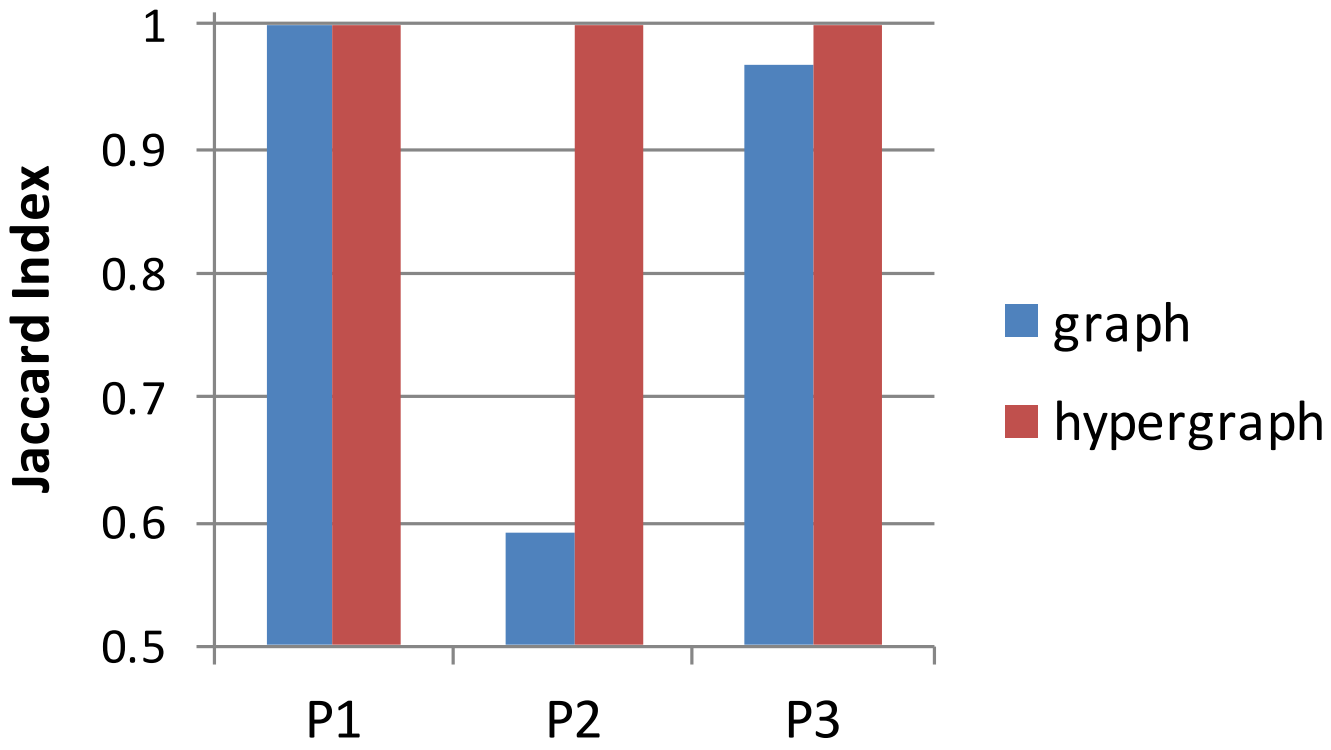
Sandia National Laboratories

	P1	P2	P3	Hyperedge Cardinality
Number of clusters	10	5	10	5
Vertices per cluster*	10,000	10,000	10,000	10,000
Intra-cluster hyperedges*	40,000	20,000	20,000	20,000
Inter-cluster hyperedges*	50,000	200,000	200,000	200,000
Intra-cluster hyperedge cardinality*	5	10	5	3-10
Inter-cluster hyperedge cardinality*	5	3	5	3-10

- Generate hypergraph incidence matrices
  - 4 different sets of parameters
  - Different levels of difficulties
  - 10 randomly generated hypergraphs for each parameter set

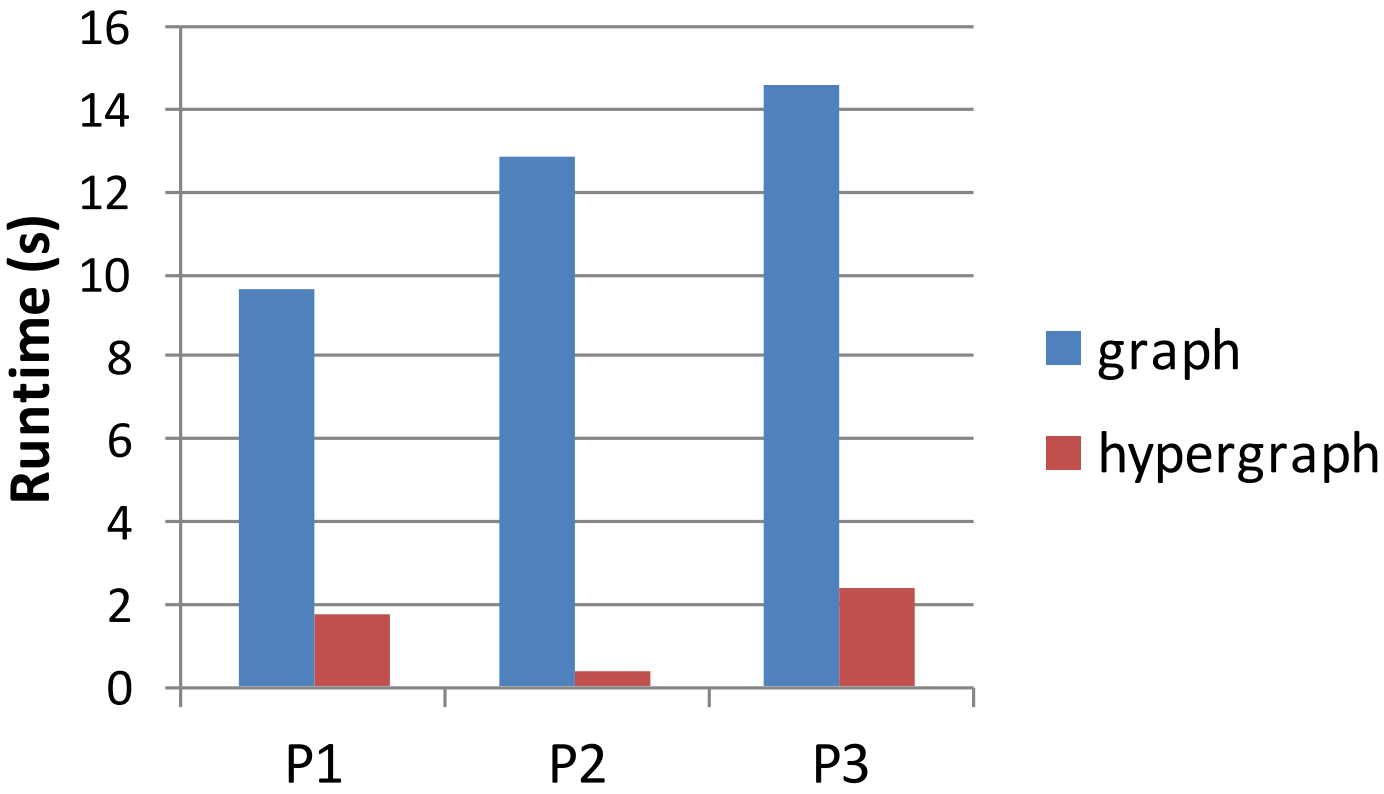
* Designates mean value

Clustering Quality: P1, P2, P3



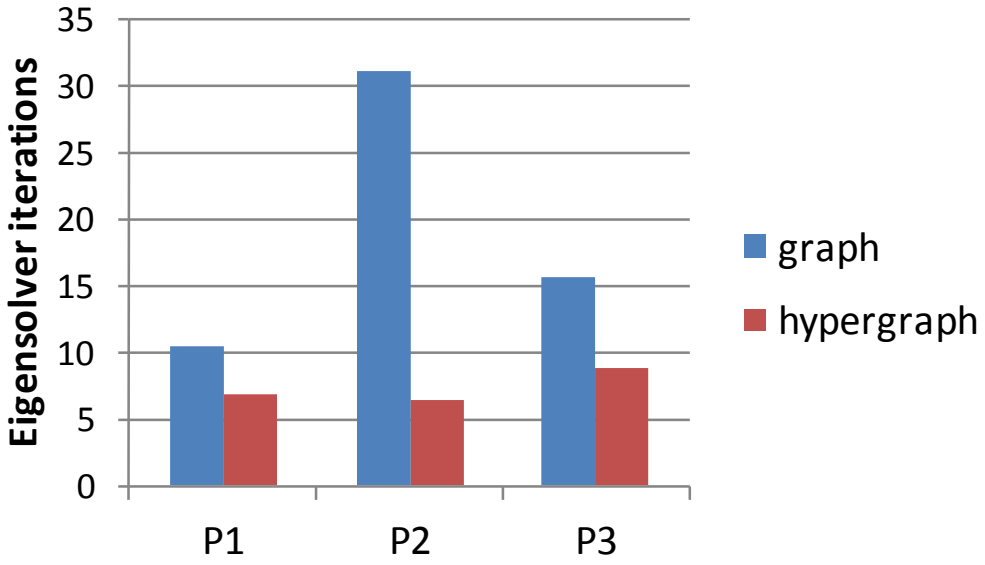
Hypergraph based clusters more similar to "ground truth" clusters than graph based clusters

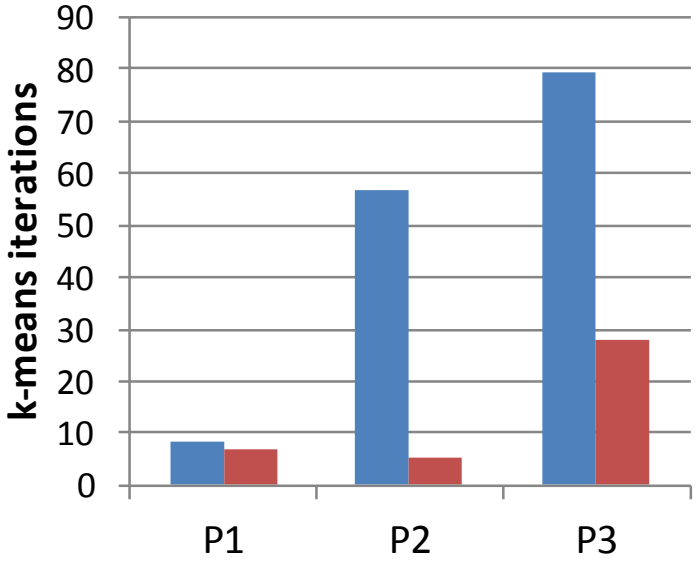
Runtimes: P1, P2, P3



Hypergraph models significantly more computationally efficient than graph models (up to 30x faster)

Iterations: P1, P2, P3

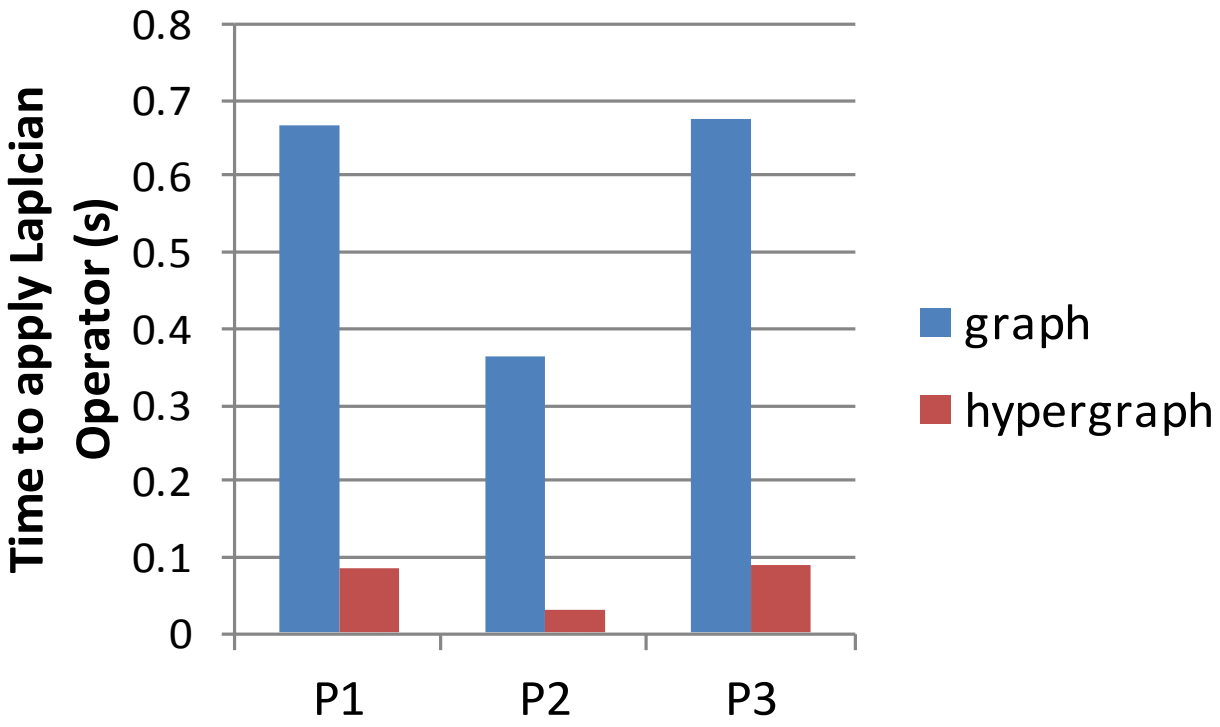




- Hypergraph required fewer LOBPCG iterations than graph
  - Better separation of eigenvalues in hypergraph Laplacian
- Hypergraph required fewer k-means iterations

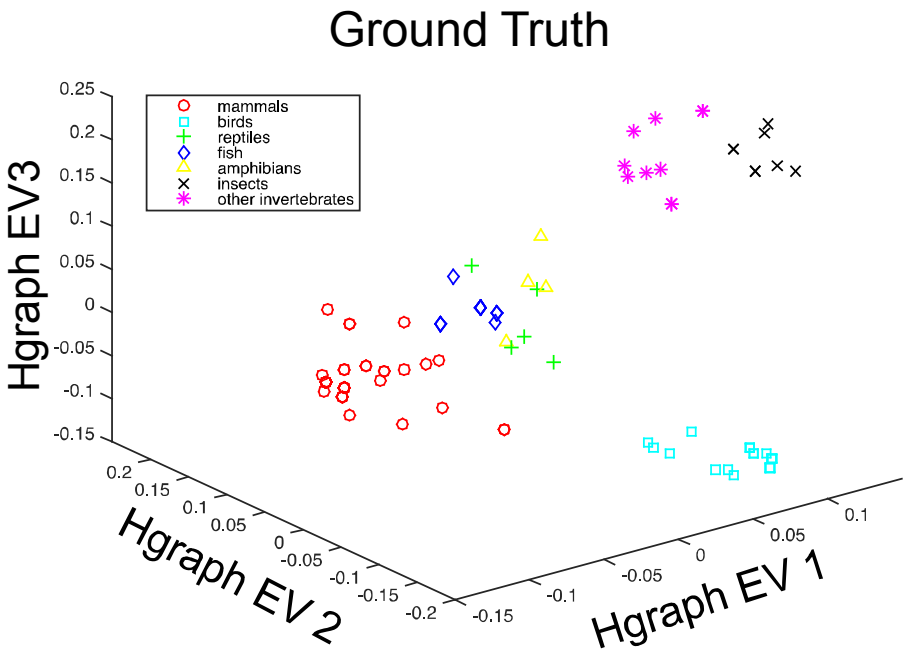
Hypergraph models converged faster than graph models

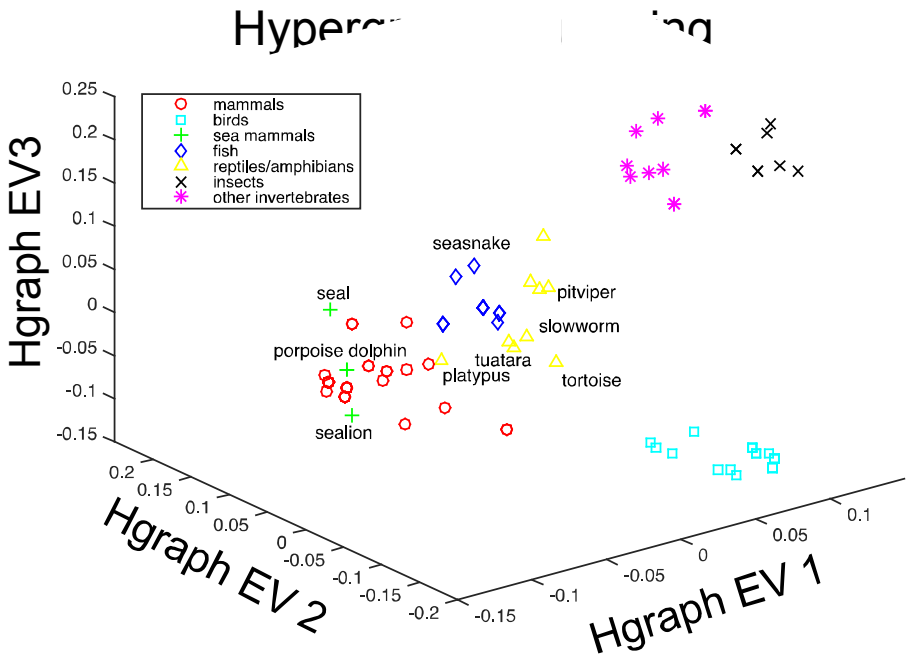
Laplacian Operator Apply: P1, P2, P3



Laplacian operator apply more efficient for hypergraph model than graph model (up to 12x faster)

Real Data – Hypergraph, 7 Clusters



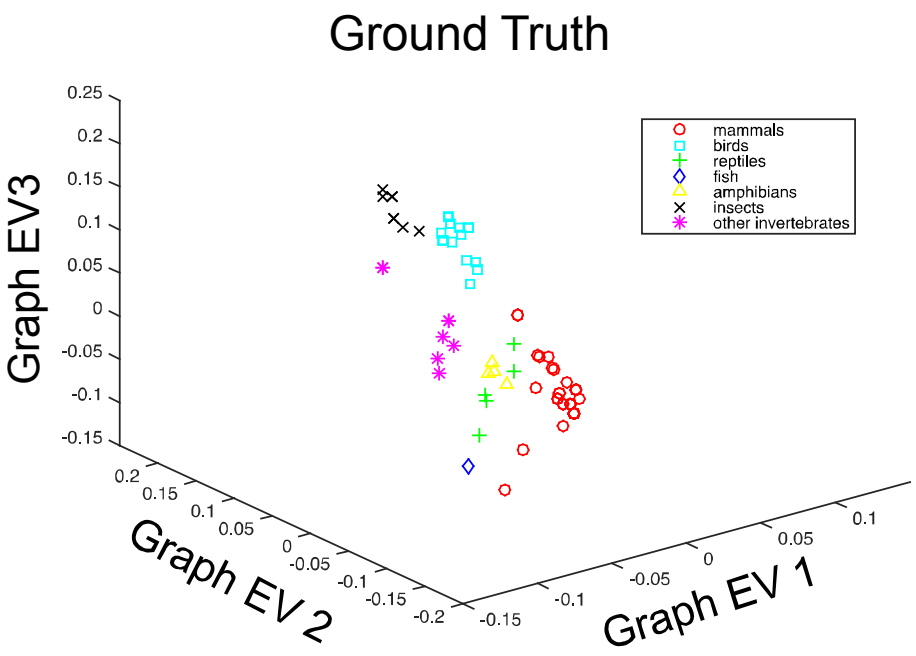


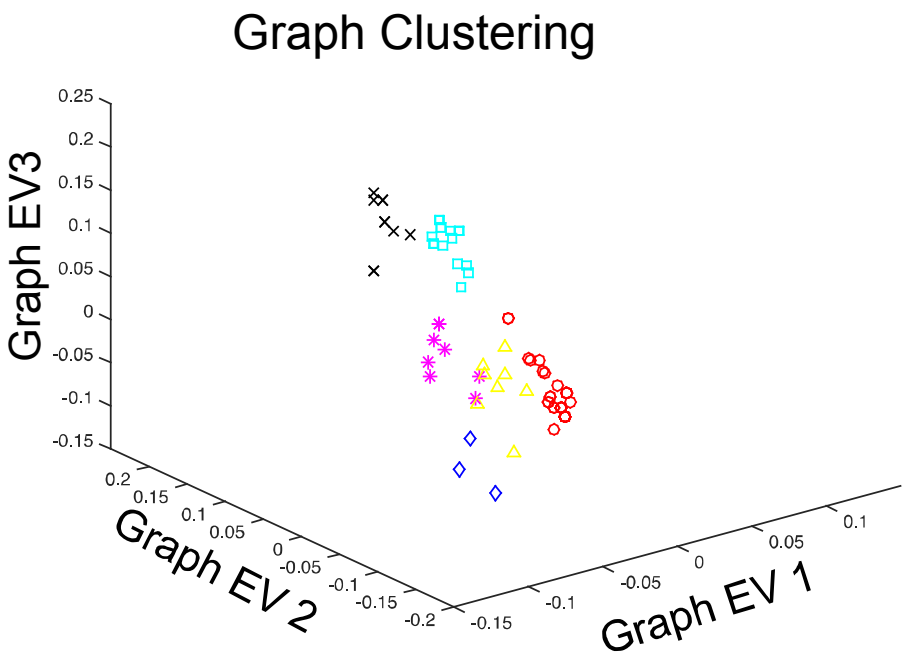
- UCI ML Repository Zoo Data Set*
  - Animals with 16 attributes (# legs, eggs, …) + category
- Hypergraph Clustering vs. ground truth for 7 clusters
  - Jaccard index: 0.815 (Graph model 0.743)
  - Merged reptiles/amphibians, new category: sea mammals

* https://archive.ics.uci.edu/ml/

EV=eigenvector

Real Data – Graph, 7 Clusters

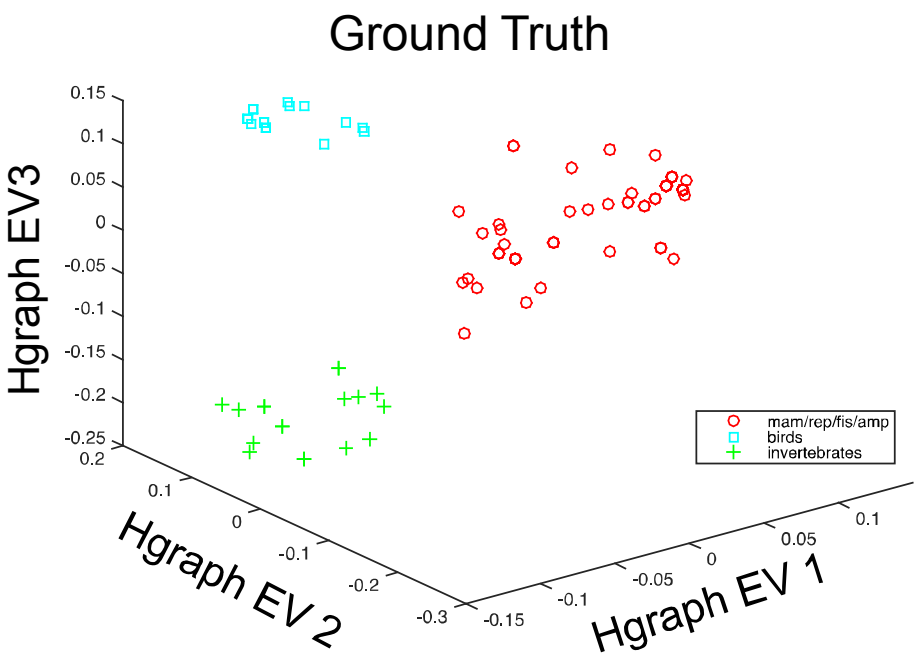


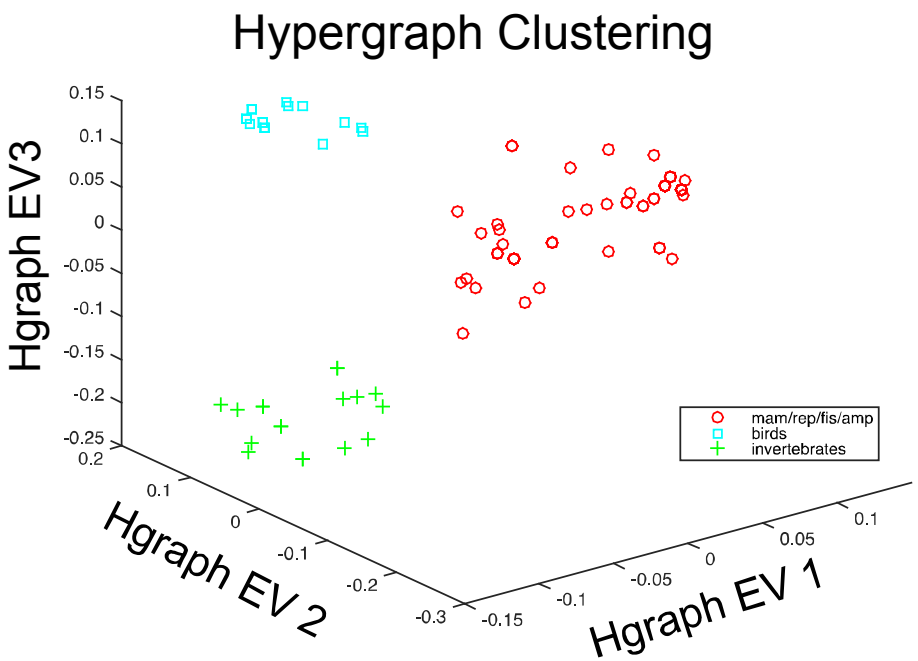


- Graph Clustering vs. ground truth for 7 clusters
  - Jaccard index: 0.743
  - Data harder for kmeans to separate
    - Insects, birds, mammals, other invertebrates, ?

EV=eigenvector

Real Data – Hypergraph, 3 Clusters



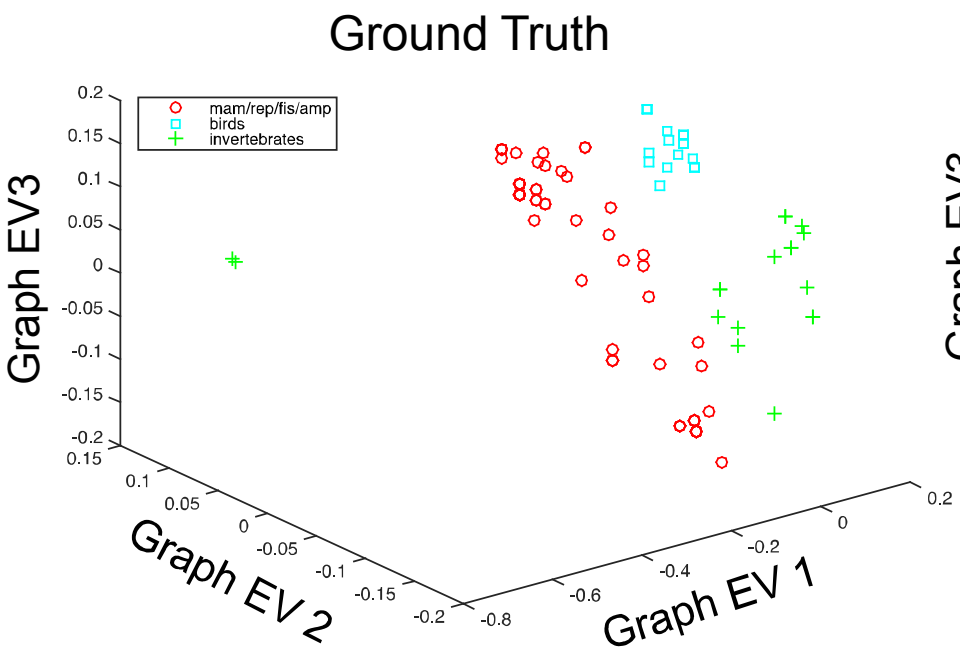


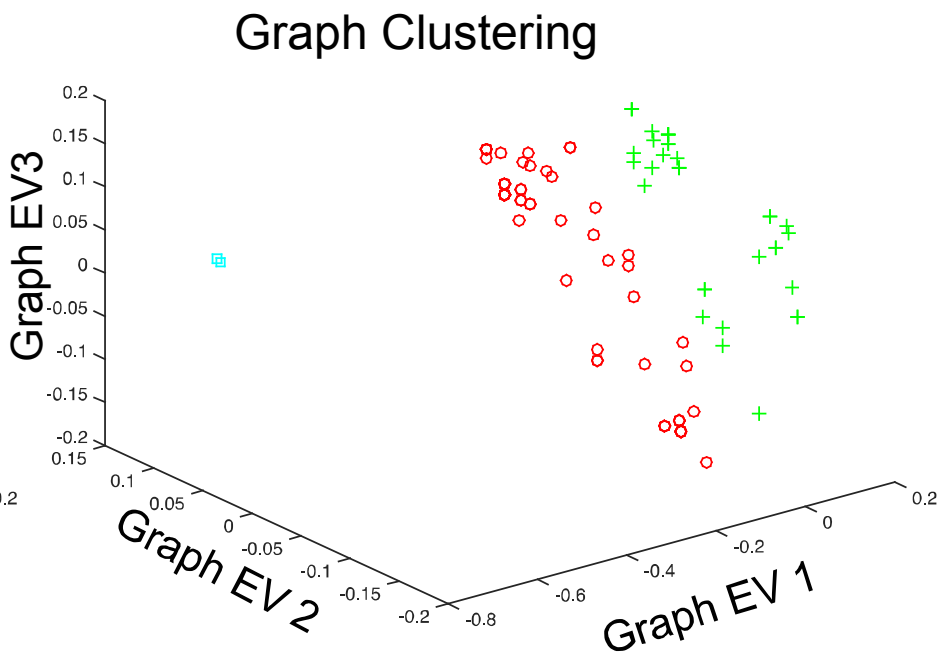
- UCI ML Repository Zoo Data Set*
  - Merged into 3 clusters: mammals/reptiles/fish/amphibians, birds, invertebrates
- Hypergraph Clustering vs. ground truth for 3 clusters
  - Jaccard index: 1.000 (Graph model 0.865)

* https://archive.ics.uci.edu/ml/

EV=eigenvector

Real Data – Graph, 3 Clusters





- Graph Clustering vs. ground truth for 3 clusters
  - Jaccard index: 0.865