SAND2016-2317C

Clustering network data through effective use of eigensolvers and hypergraph models

Alicia Klinvex, Michael Wolf, and Daniel Dunlavy

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. 2011-XXXXP

Motivating problem: Community detection

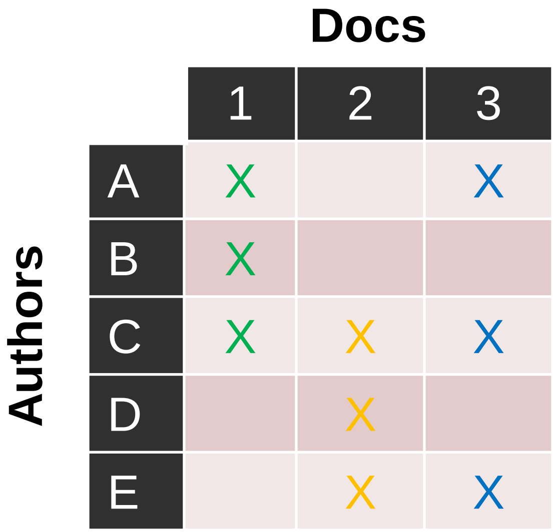
- Determine groupings of data objects given sets of relationships amongst those objects
- Relationships may be represented in a graph or hypergraph
  - Graphs represent pairwise relationships
  - Hypergraphs represent relationships among groups of things
- Applications
  - Finding emerging research trends 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)

Outline

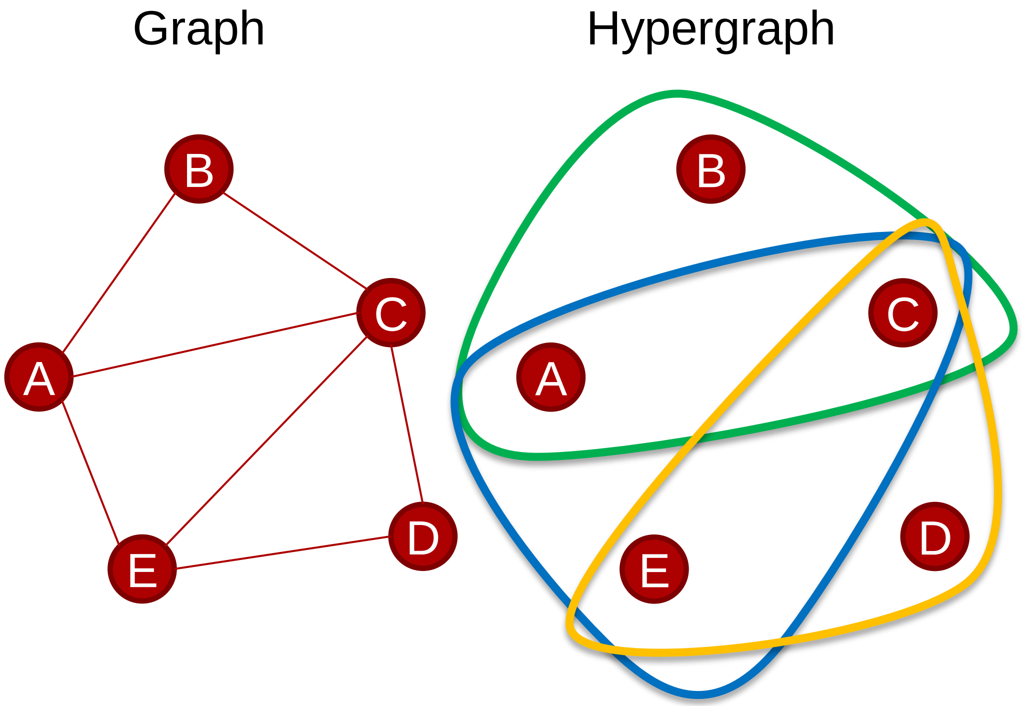
- Introduction to hypergraphs
- Description of spectral clustering algorithm
- Exploration of eigenvalue problems occurring in spectral clustering
- Spectral clustering results

- Explore the usage of hypergraphs to model relational data
- Understand how to effectively use eigensolvers in spectral analysis of this data

What is a hypergraph?



	Docs		
Authors	1	2	3
A	X		X
B	X		
C	X	X	X
D		X	
E		X	X

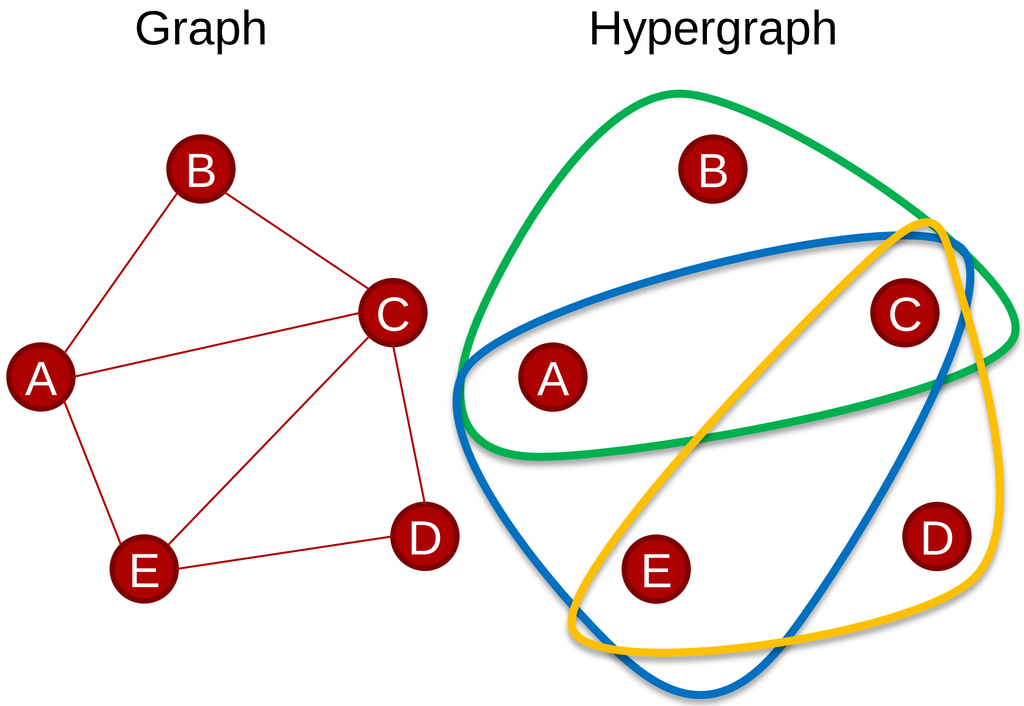


- Generalization of graph
  - Hyperedges represent multiway relationships between vertices
  - A hyperedge is a set of vertices of arbitrary size
  - Hyperedges can connect more than 2 vertices

What is a hypergraph?

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

Hypergraph incidence matrix



- Multiway relationships can be represented nonambiguously
  - Did A, B, and C write a paper together?
- Relational data is hypergraph incidence matrix
  - We can convert a hypergraph to a graph via clique expansion

Hypergraph clique expansion

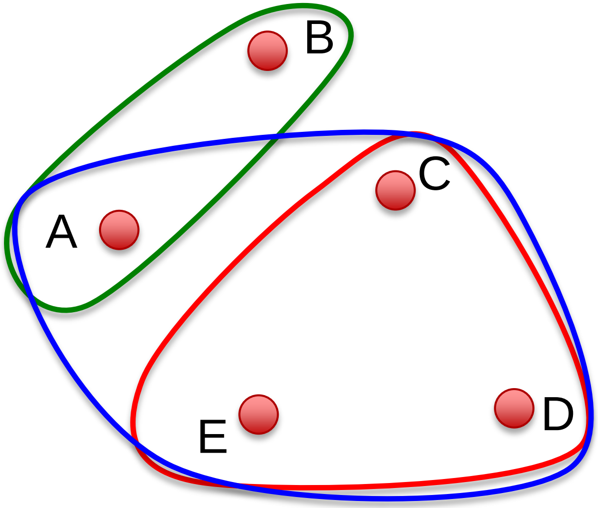
Hyperedges

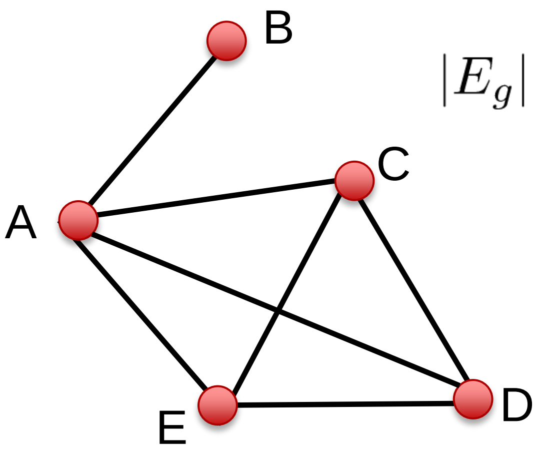
Vertices

	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	X				X	X	X			
B	X									
C		X	X		X			X	X	
D		X		X		X		X		X
E			X	X			X		X	X





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

Weighted hypergraph clique expansion

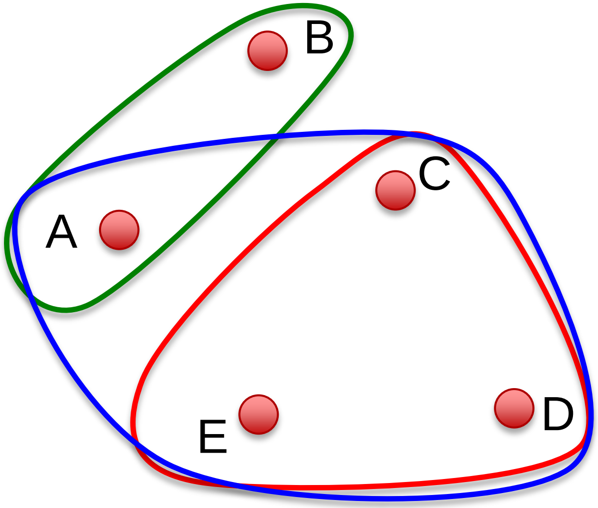
Hyperedges

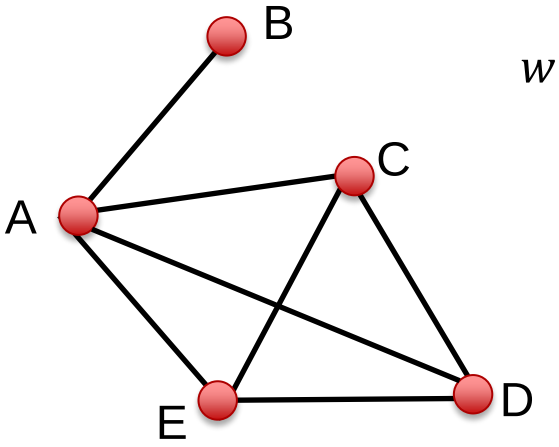
Vertices

	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





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

Unifying expansions and Laplacians: Agarwal et al., 2006

Computational advantages of hypergraphs

1		1
1		
	1	1
	1	1
	1	1

1				⅓	⅓	⅓			
1									
	½	½		⅓			⅓	⅓	
	½		½		⅓		⅓		⅓
		½	½			⅓		⅓	⅓

Hypergraph incidence matrix

Graph Incidence matrix

- Hypergraphs require significantly less storage space than graphs generated using clique expansion

$$|E_g| = \sum_{e\in E_h} \left(\begin{array}{c} d(e) \\ 2 \end{array} \right)$$

- Hypergraphs require fewer operations for a matrix-vector multiplication

How do we detect communities in graphs and hypergraphs?

- Spectral clustering (Ng, et al., 2002)
  - Compute the smallest eigenpairs of the normalized graph or hypergraph Laplacian (Zhou, et al., 2006)

$$L_G = I - {D_v}^{-1/2}\left(H_g {H_g}^T - D_v\right){D_v}^{-1/2}$$

$$L_H = I - {D_v}^{-1/2} H_h {D_e}^{-1} {H_h}^T {D_v}^{-1/2}$$

    - Laplacian is never explicitly formed

1		1
1		
	1	1
	1	1
	1	1

1				⅓	⅓	⅓			
1									
	½	½		⅓			⅓	⅓	
	½		½		⅓		⅓		⅓
		½	½			⅓		⅓	⅓

How do we detect communities in graphs and hypergraphs?

- Spectral clustering (Ng, et al., 2002)
  - Compute the smallest eigenpairs of the normalized graph or hypergraph Laplacian (Zhou, et al., 2006)

$$L_G = I - {D_v}^{-1/2}({H_g H_g}^T - D_v){D_v}^{-1/2}$$

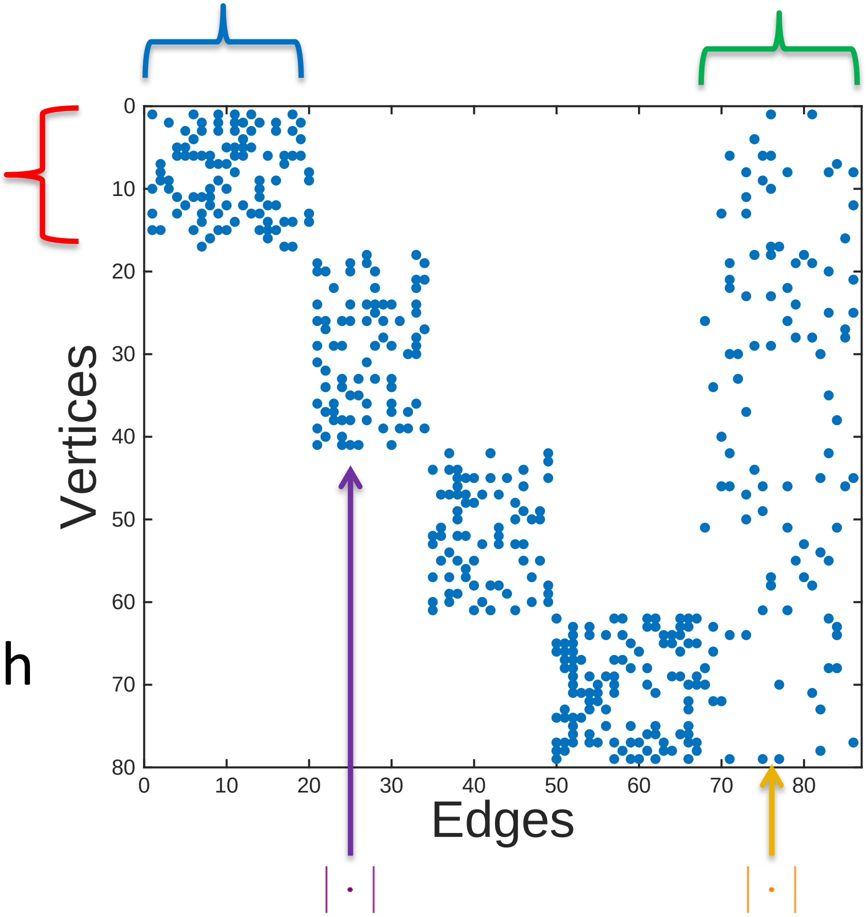
$$L_H = I - {D_v}^{-1/2} H_h {D_e}^{-1} {H_h}^T {D_v}^{-1/2}$$

  - Perform k-means clustering on those eigenvectors
    - Partition a set of observations into clusters in which each observation belongs to the cluster with the nearest mean
- Quality of our results is measured using the Jaccard index
  - T = true cluster assignments
  - P = predicted cluster assignments

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

Randomly Generated Hypergraphs

- Parameters
  - Clusters
  - Nodes per cluster
  - Intra-cluster hyperedges
  - Inter-cluster hyperedges
  - Hyperedge cardinalities
    - Intra-cluster
    - Inter-cluster
- We also generate a ground truth clustering vector



Incidence Matrix: H_h

Solving eigenvalue problems

- Which eigensolver should we use?
- How many eigenpairs should we compute?
- How accurate do the eigenpairs need to be?

Choice of eigensolver

- We use the eigensolvers available in Trilinos/Anasazi
  - Locally Optimal Block Preconditioned Conjugate Gradient method (LOBPCG)
  - TraceMin-Davidson
  - Riemannian Trust Region method (RTR)

Experimental results

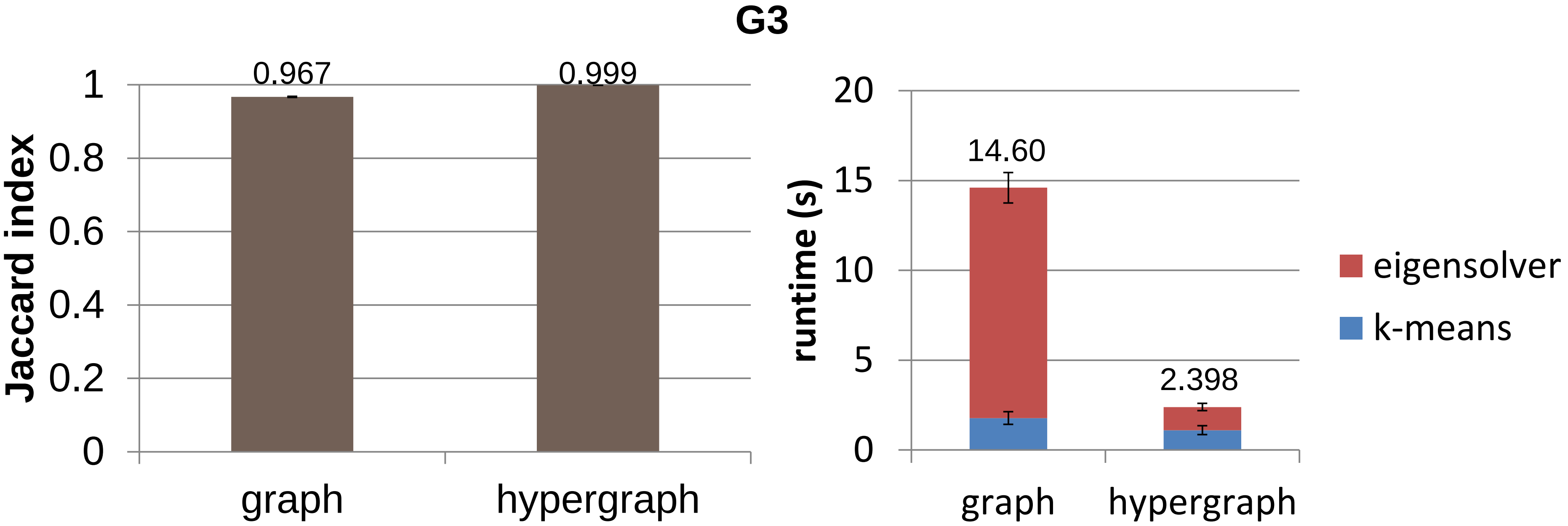
- Experiments were conducted on a 24 core machine with 128 GB of memory using 16 MPI processes
- Runtime parameters
  - 10 matrices of each type

	G1	G2	G3
Number of clusters	10	5	10
Nodes per cluster	10,000	10,000	10,000
Intra/Inter-cluster hyperedges	40,000 / 50,000	20,000 / 200,000	20,000 / 200,000
Intra/Inter-cluster h-edge cardinality	5 / 5	10 / 3	5 / 5

  - 5 k-means trials per matrix
  - Eigensolver: LOBPCG
  - Number of computed eigenpairs: same as number of clusters*
  - Tolerance: 1e-3*

*unless otherwise stated

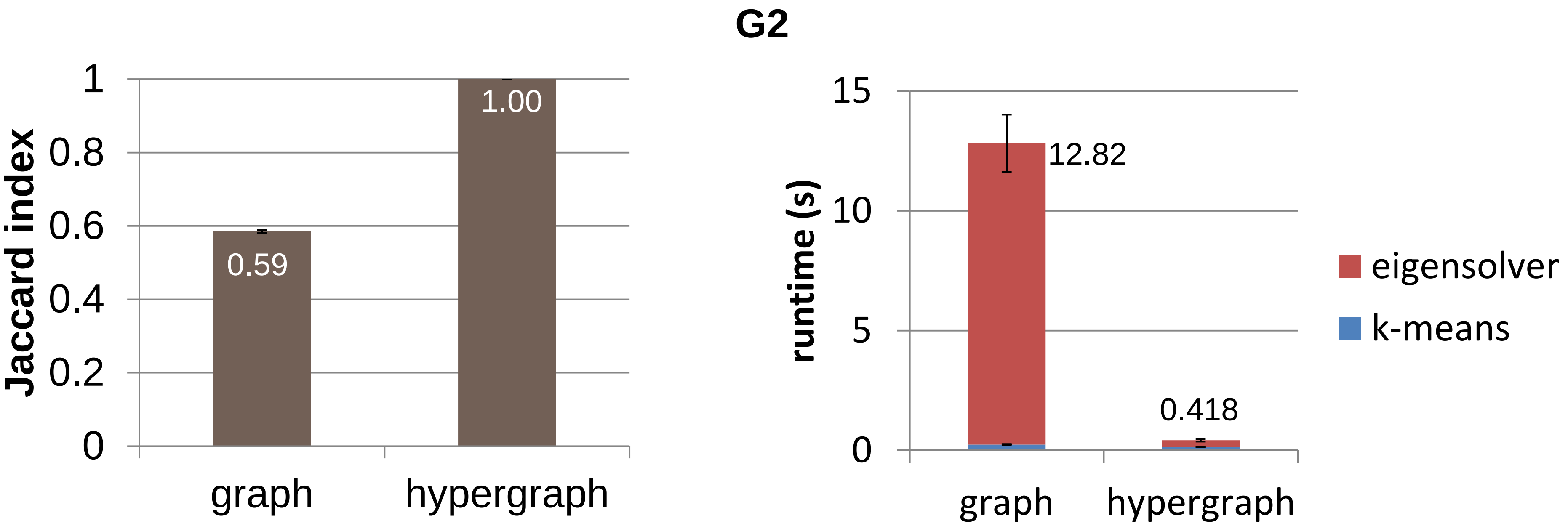
How do graph and hypergraph results compare?


G3

	Graph	Hypergraph
k-means iterations	79.4	28.1
LOBPCG iterations	15.6	8.9

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

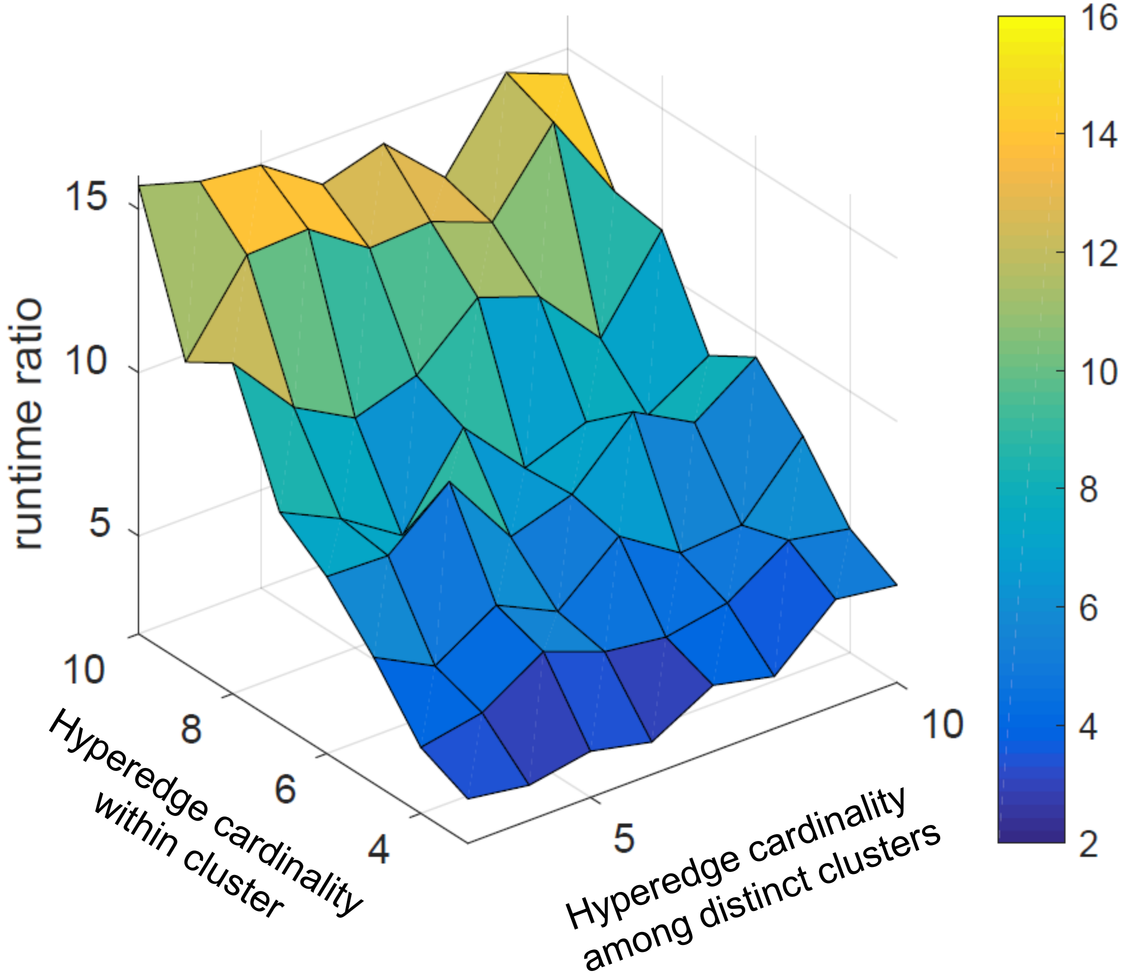
How do graph and hypergraph results compare?



	Graph	Hypergraph
k-means iterations	56.8	5.4
LOBPCG iterations	31.1	6.5

Number of clusters	5
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	20,000 200,000
Intra/Inter-cluster h-edge cardinality	10 3

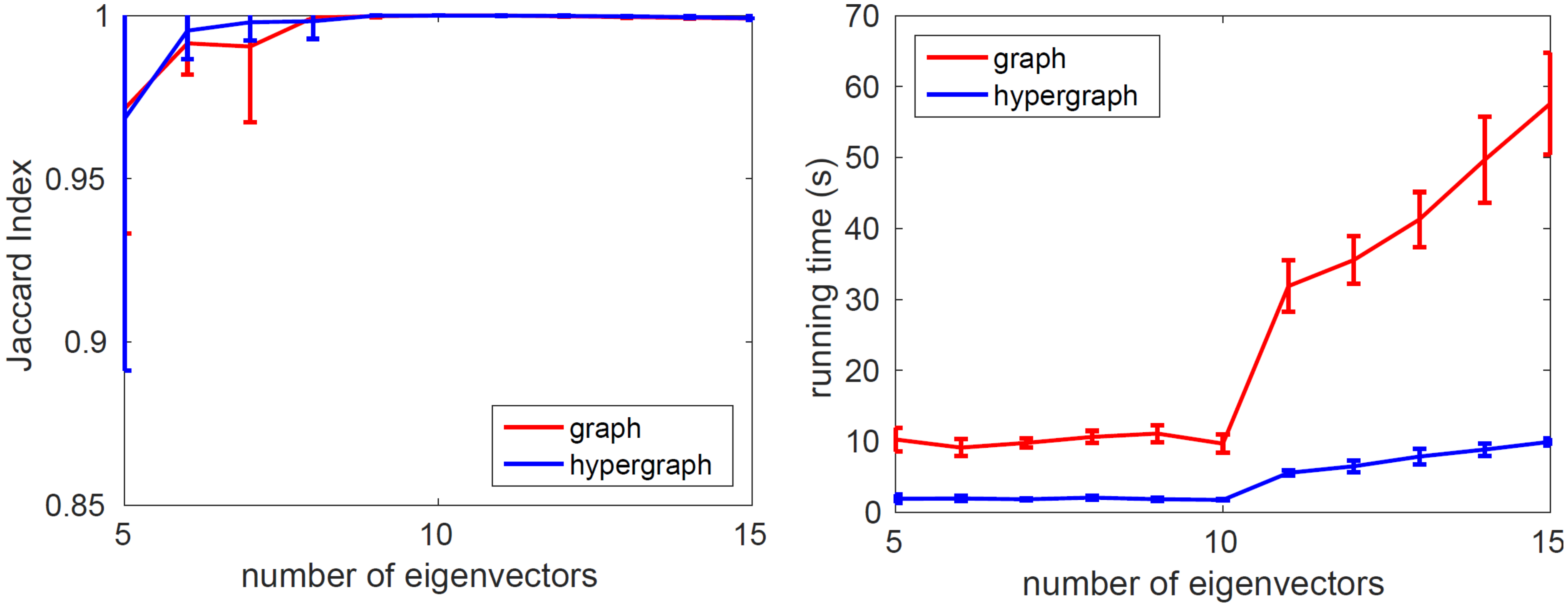
How do graph and hypergraph runtimes compare?



Number of clusters	5
nodes per cluster	10,000
Intra/Inter-cluster hyperedges	40,000 50,000

Jaccard index was always 1

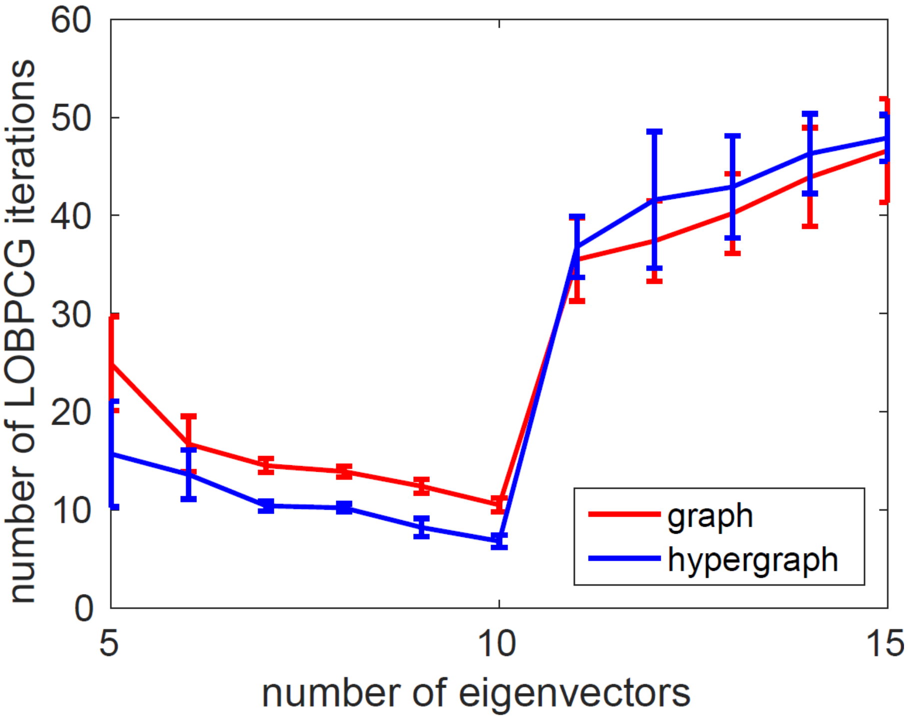
How many eigenvectors should we calculate?



Less noisy data: G1

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

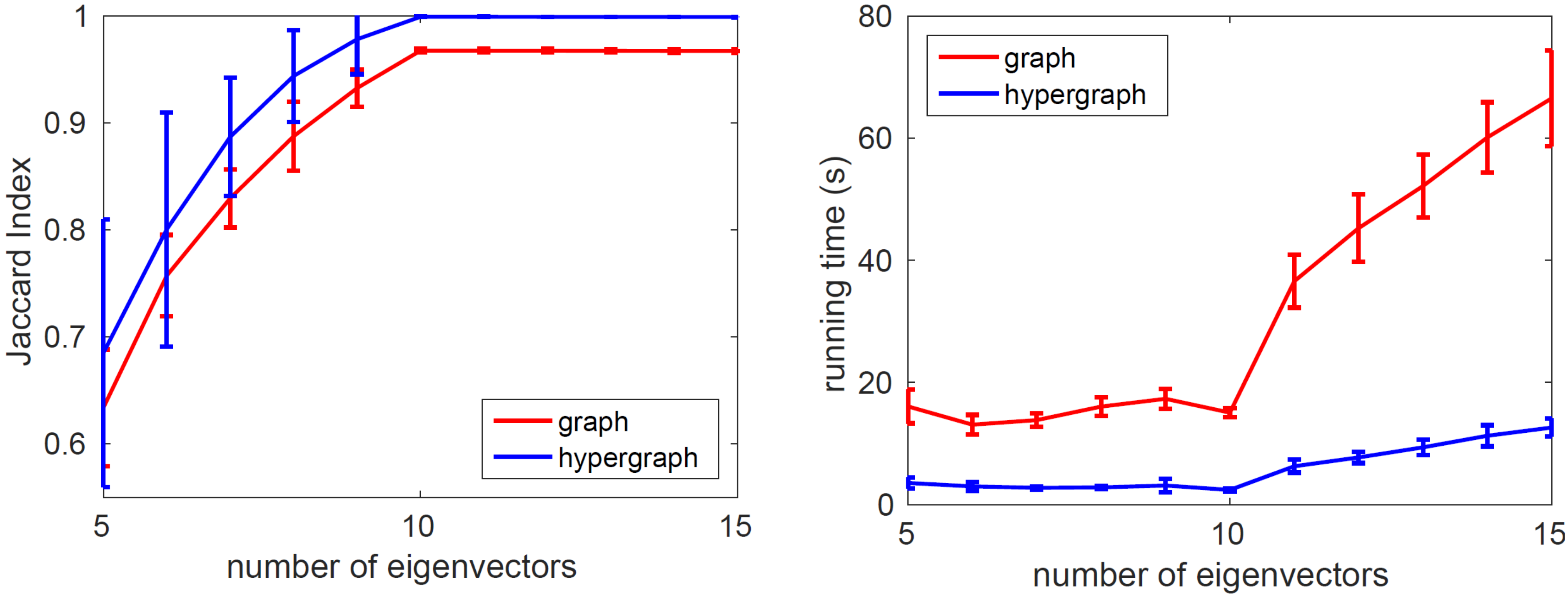
How many eigenvectors should we calculate?



Less noisy data: G1

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

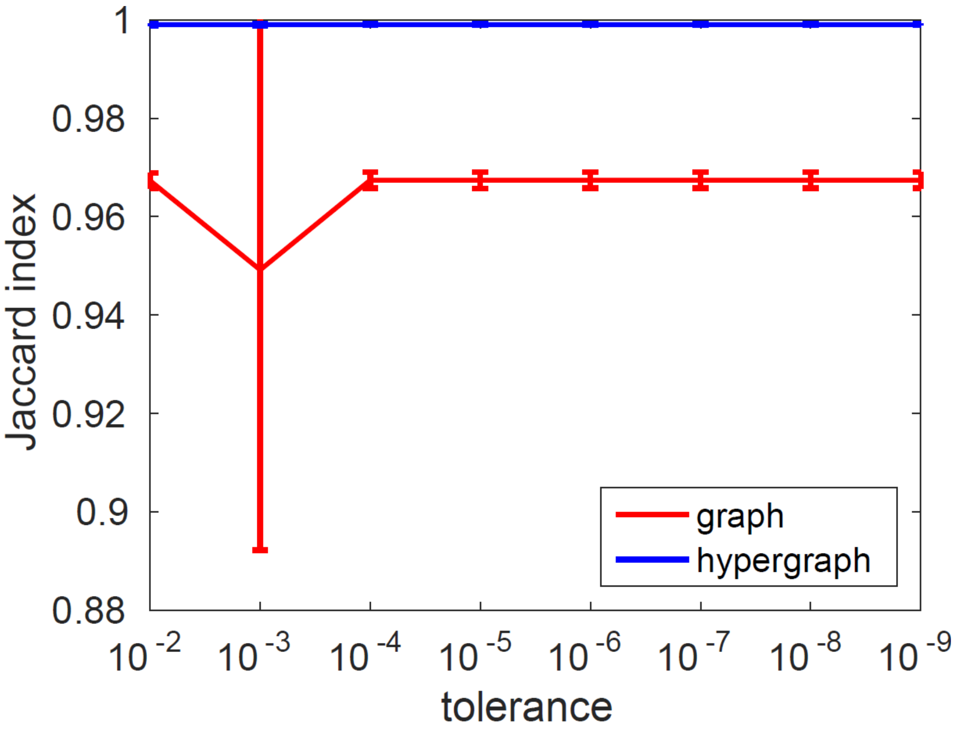
How many eigenvectors should we calculate?

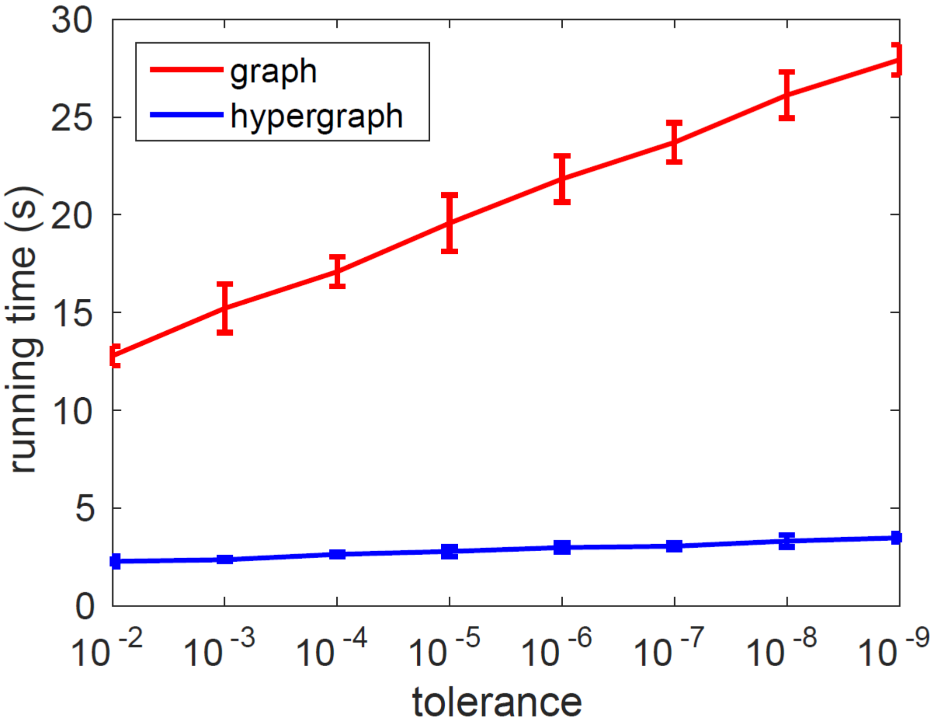


Noisier data: G3

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?





G3

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

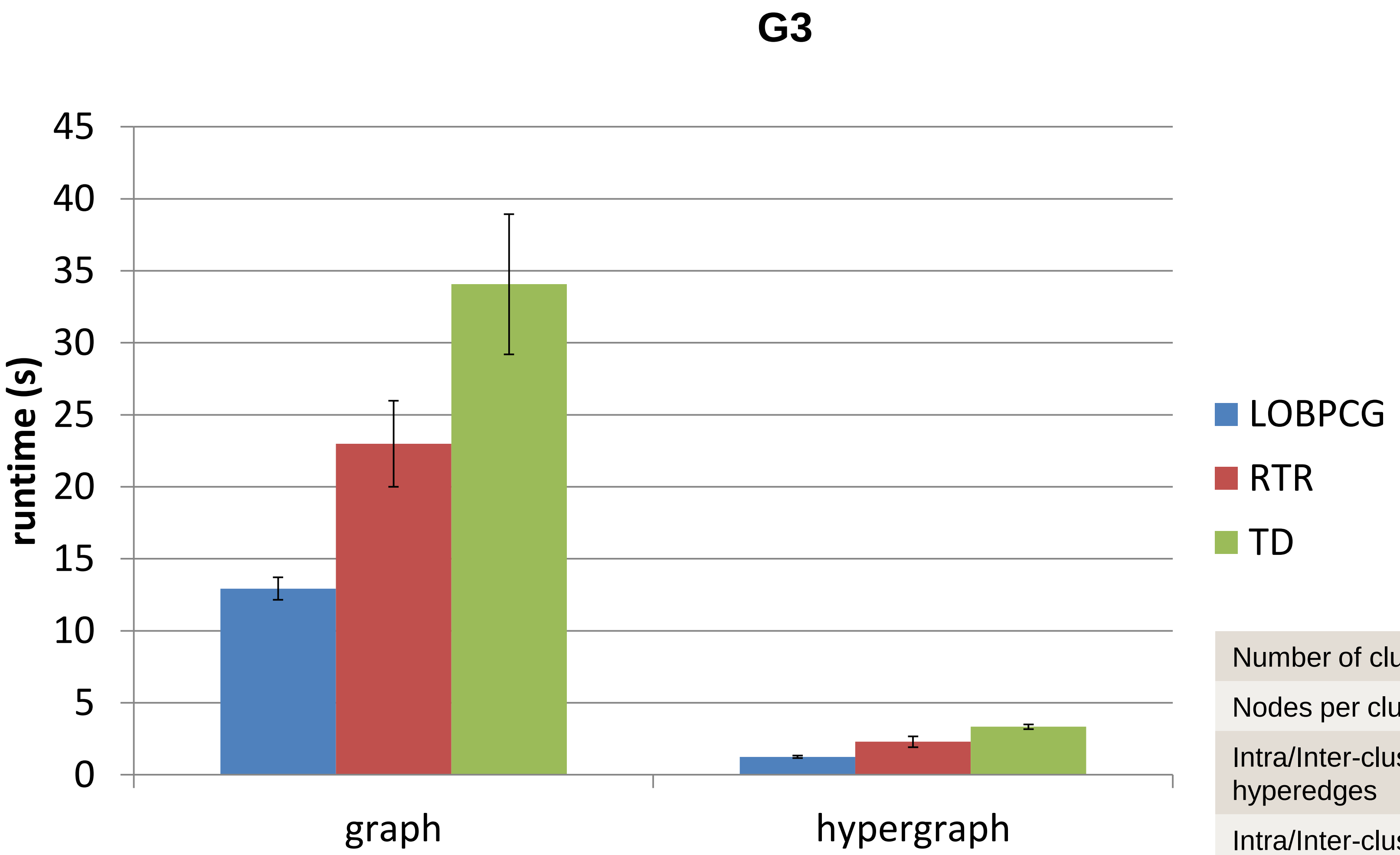
eigenvectors for tolerance of 10^{-1} were random

Conclusions

- Graph vs hypergraph
  - Preliminary results suggest a **dramatic runtime difference** between eigensolver computation for graph and hypergraph case
  - **Larger Jaccard indices** for hypergraph over graph for several problem classes
- Eigensolver
  - Low tolerances are acceptable
  - Choice of number of eigenvectors is very important
  - LOBPCG is effective for problems we studied
- Currently exploring real world problems where hypergraphs may be a better choice

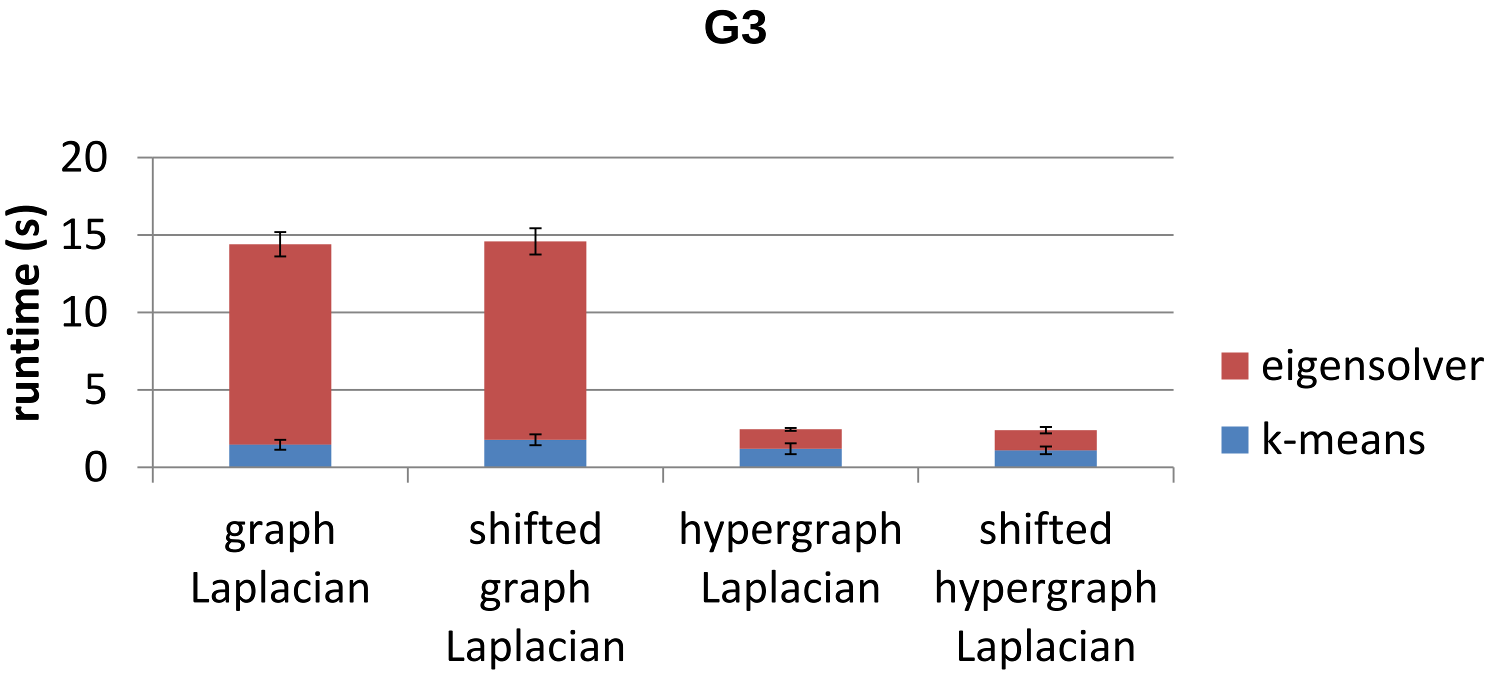
BACKUP SLIDES

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

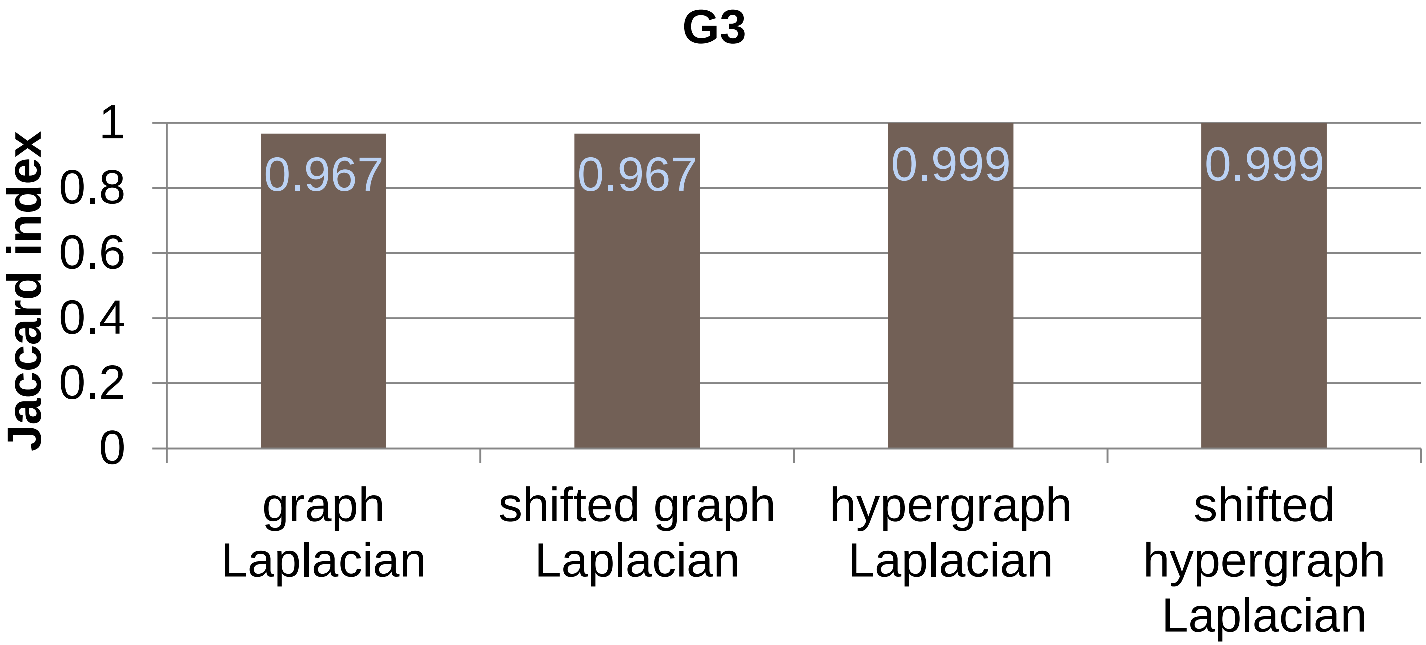
Should we compute the eigenpairs of the Laplacian or the shifted Laplacian?



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

	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

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



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

	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

Formulation of the eigenvalue problem

- Computing the smallest eigenpairs of

$$L_g = I - D_v{}^{-1/2}\left(H_g H_g{}^T - D_v\right) D_v{}^{-1/2} \qquad L_h = I - D_v{}^{-1/2} H_h D_e{}^{-1} H_h{}^T D_v{}^{-1/2}$$

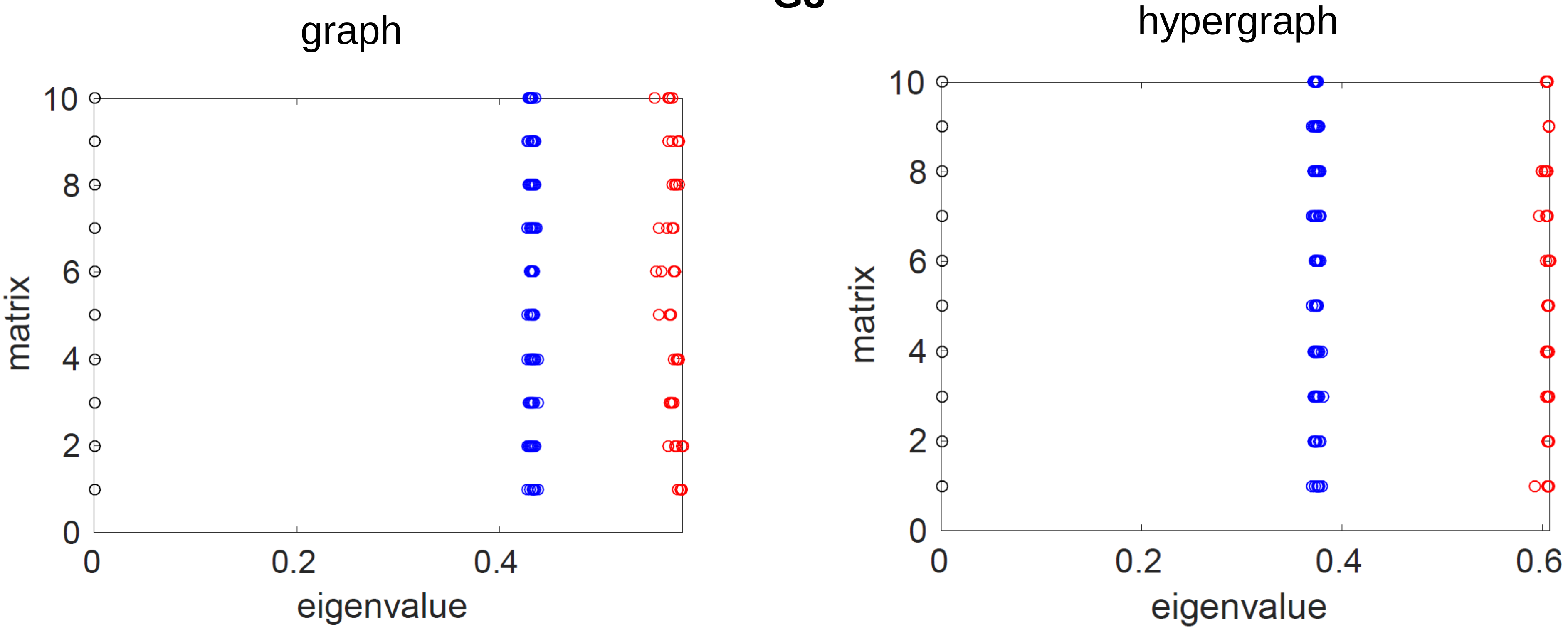
is equivalent to computing the largest eigenpairs of the shifted Laplacians

$$S_g = D_v{}^{-1/2}\left(H_g H_g{}^T - D_v\right) D_v{}^{-1/2} \qquad S_h = D_v{}^{-1/2} H_h D_e{}^{-1} H_h{}^T D_v{}^{-1/2}$$

  - 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

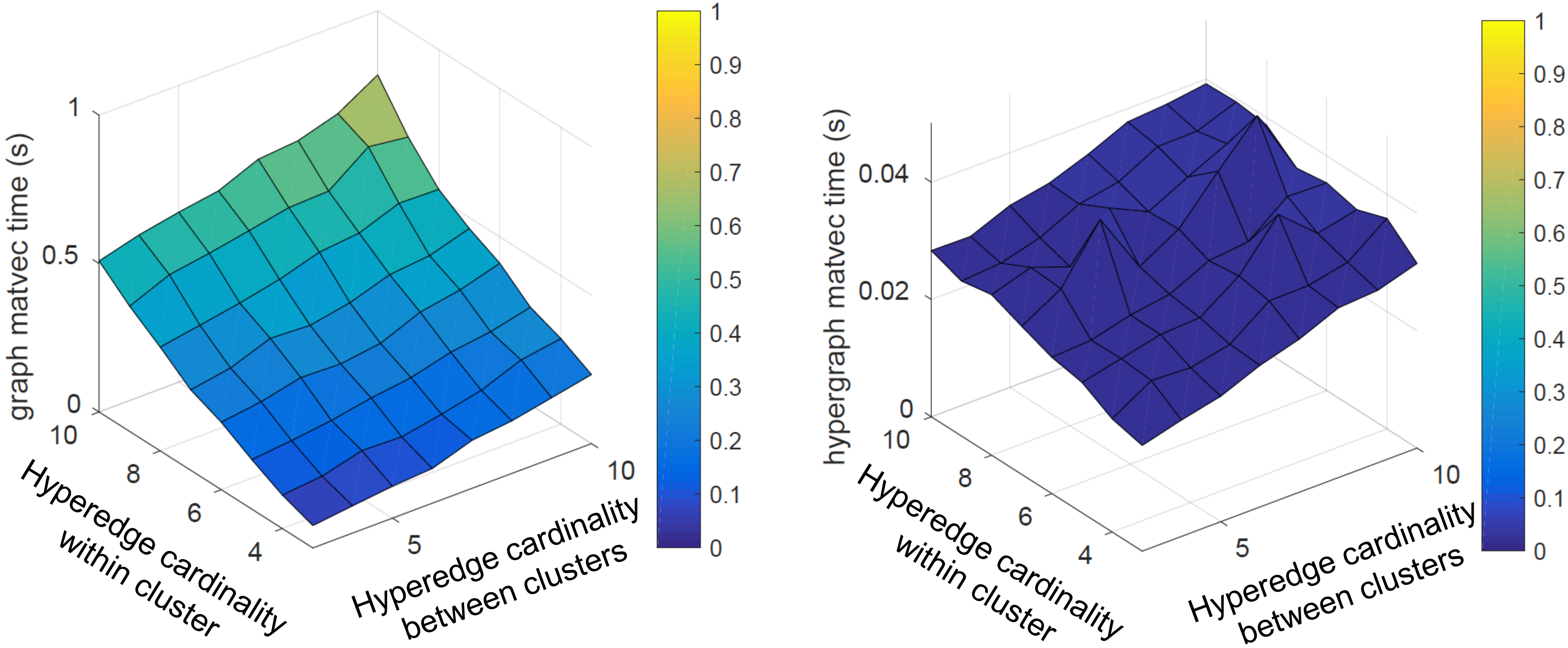
Plot of eigenvalues

G3



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

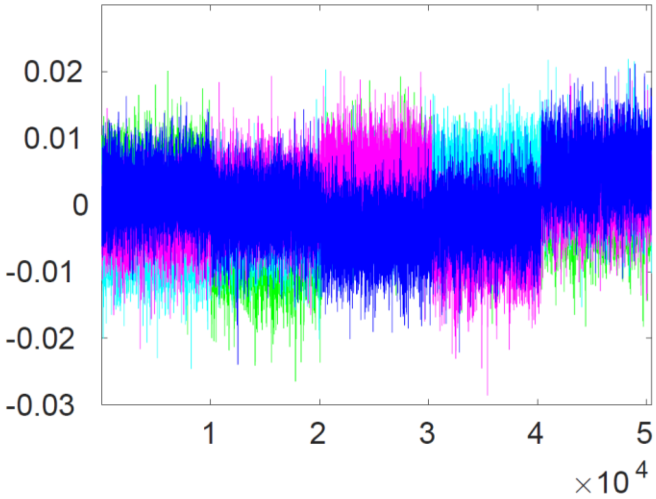
Matrix-vector multiply runtime



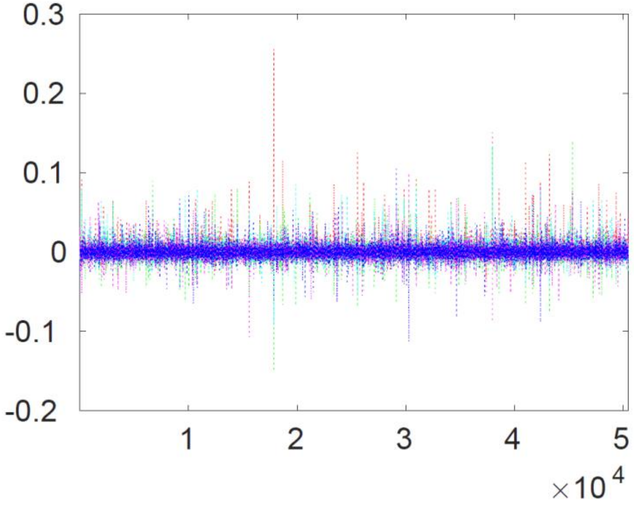
Number of clusters	5
nodes per cluster	10,000
Intra/Inter-cluster hyperedges	40,000 50,000

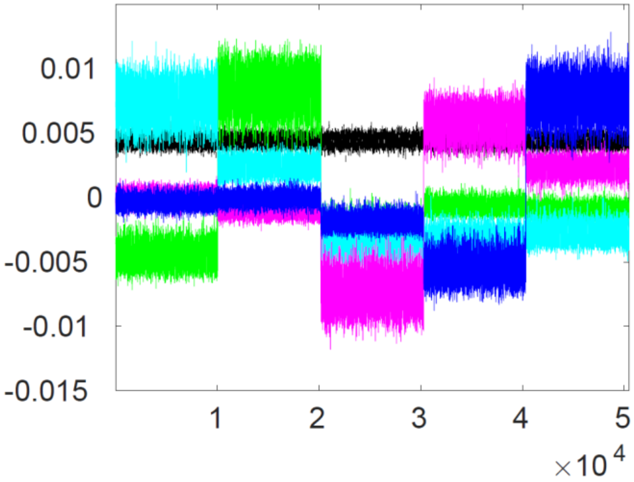
Plot of eigenvectors

G2

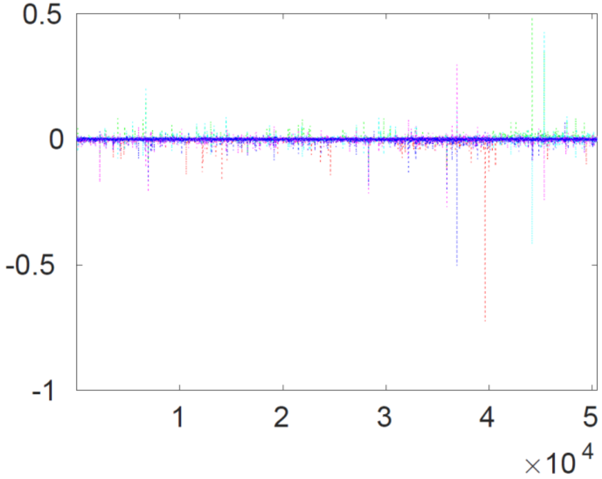


graph





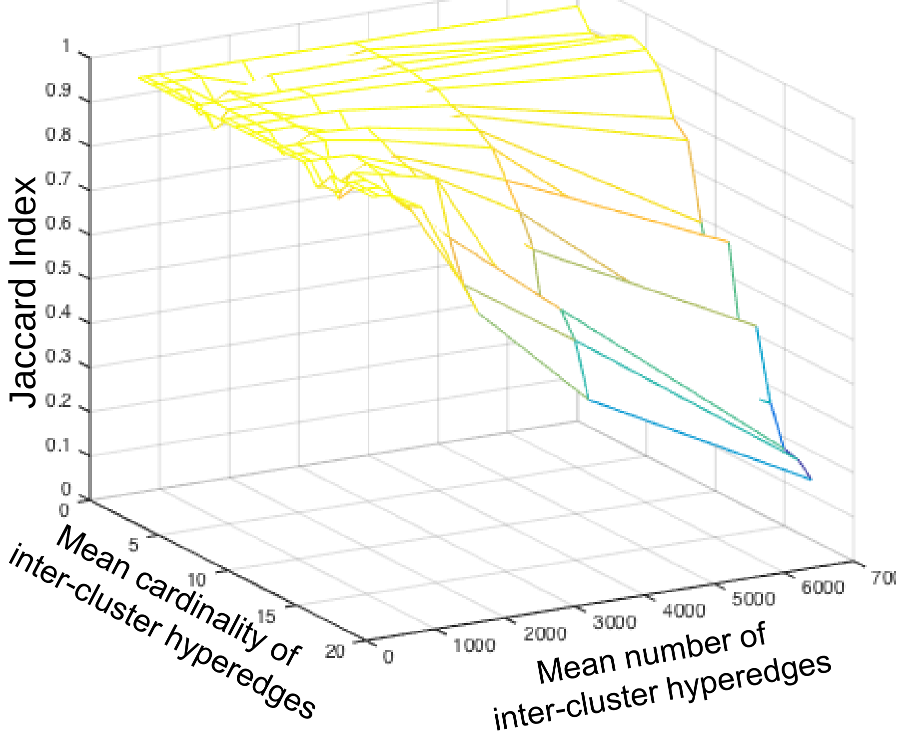
hypergraph



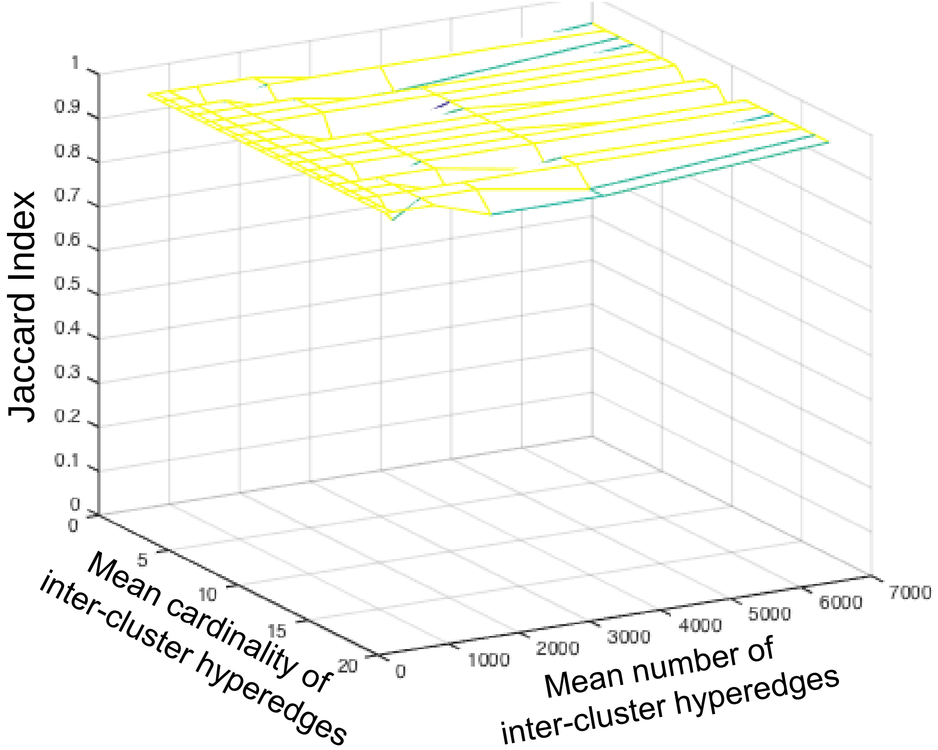
Number of clusters	5
Nodes per cluster	10,000
Intra/Inter-cluster hyperedges	20,000 200,000
Intra/Inter-cluster h-edge cardinality	10 3

Unweighted graph vs hypergraph clustering

Graph Approach

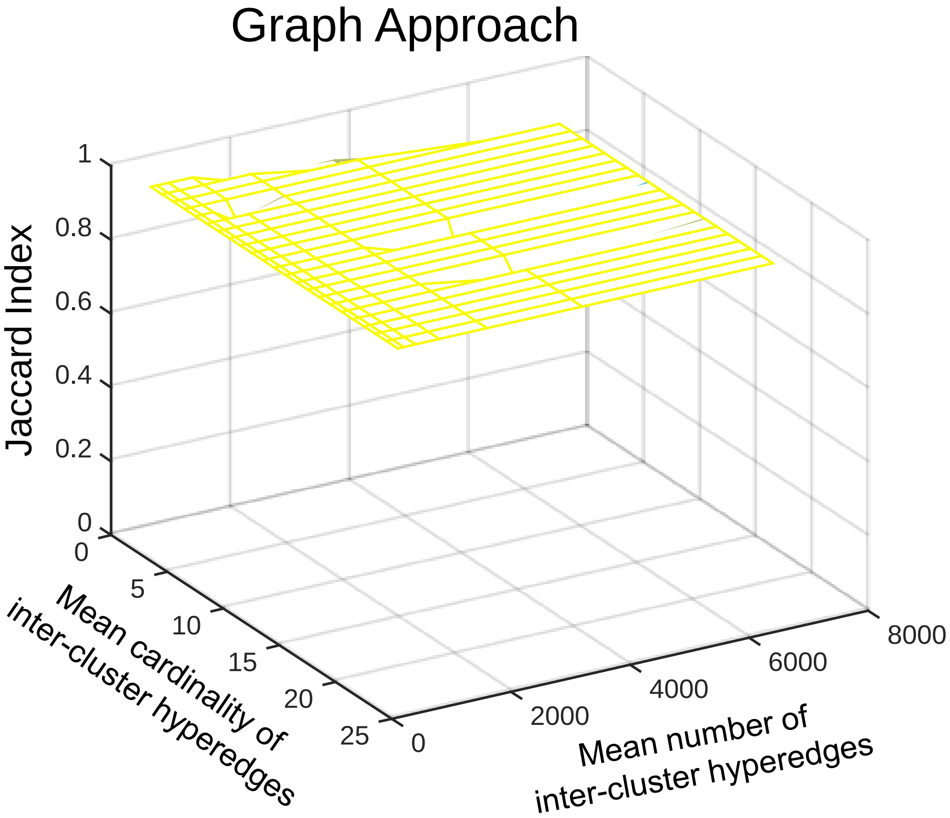


Hypergraph Approach

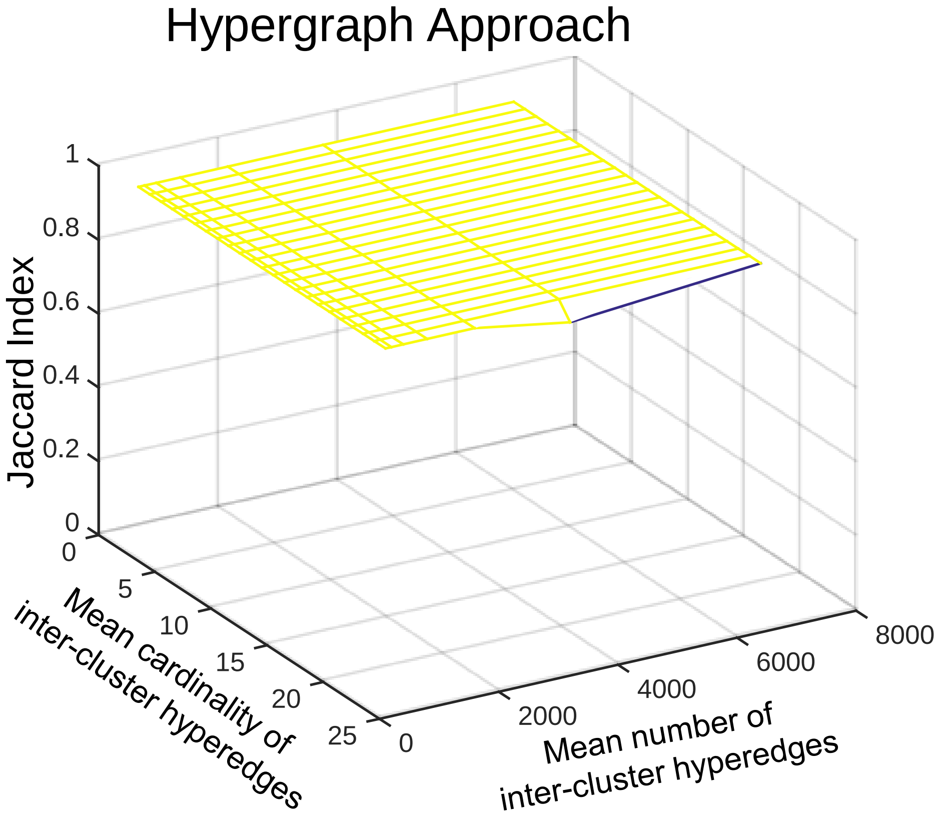


Mean intra-cluster cardinality = 3
Number of clusters 4, mean size 100
Mean number of inter-cluster edges: 800

Weighted graph vs hypergraph clustering



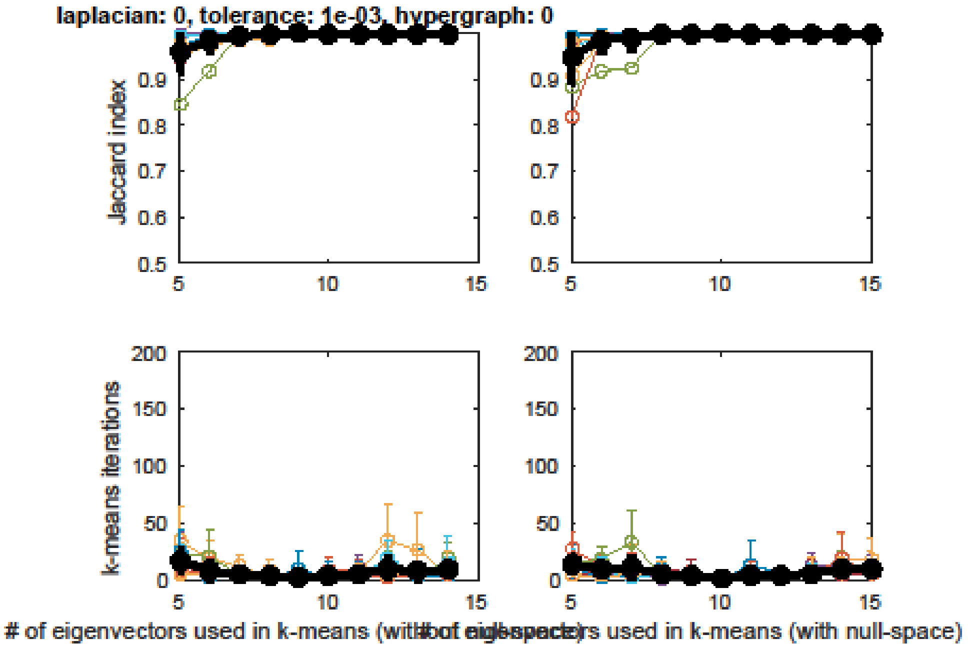
Graph Approach



Hypergraph Approach

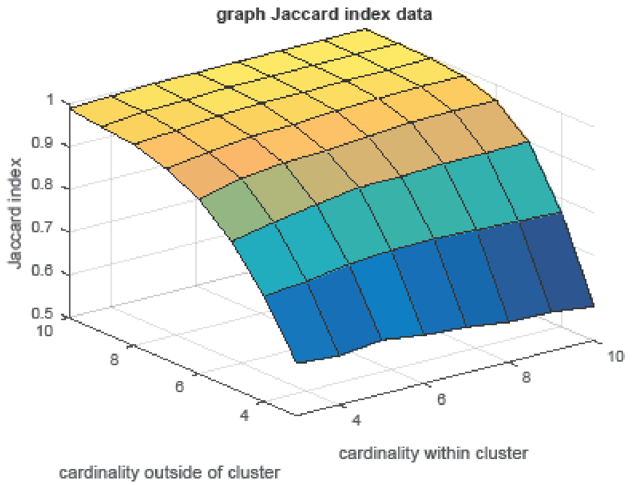
Number of clusters = 4
Mean number of vertices = 100
Number of eigenvectors = 5

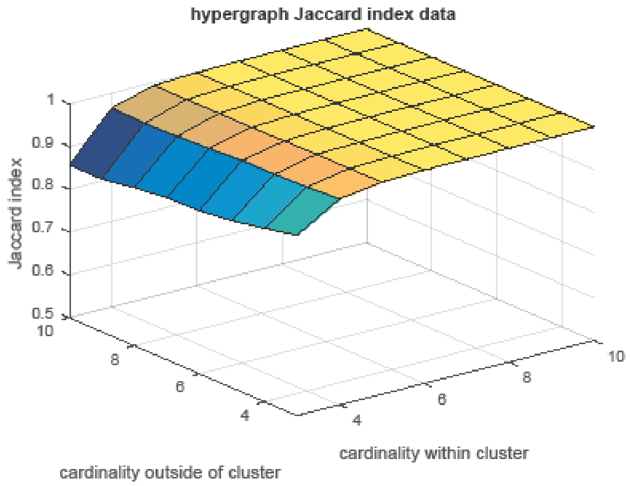
Should the null space be provided to k-means?



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

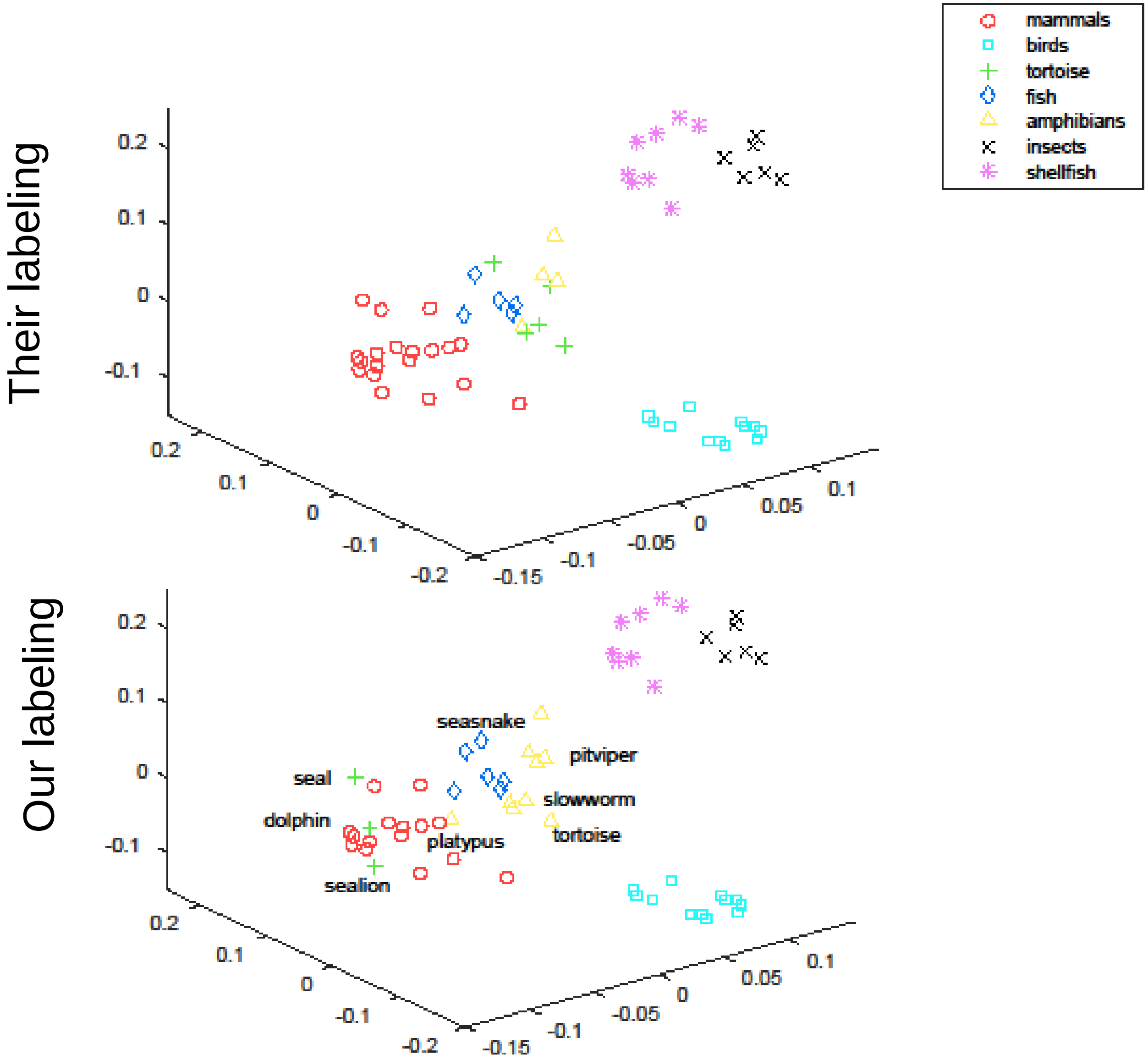
Graph vs hypergraph results





- nClusters: 5
- nVerts: 10,000
- nEdges in cluster: 20,000
- nEdges between clusters: 200,000

A real data set: zoo



Jaccard index: .815