

Computational Frameworks for Higher-order Network Data Analysis

Austin R. Benson · Cornell University

Texas A&M Institute of Data Science · October 23, 2020

Slides. bit.ly/arb-tamu-20

Graph or network data modeling important complex systems are everywhere.



Communications

nodes are people/accounts
edges show info. exchange



Physical proximity

nodes are people
edges link those that interact
in close proximity



Commerce

nodes are products
edges link co-purchased
products



Drug compounds

nodes are substances
edge between substances that
appear in the same drug

Frequently bought together



Total price: **\$55.96**

[Add all three to Cart](#)

[Add all three to List](#)

- ✓ **This item:** 6-Pack LED Dimmable Edison Light Bulbs 40W Equivalent Vintage Light Bulb, 2200K-2400K Wa
- ✓ Edison Light Bulbs, DOREShop 40Watt Antique Vintage Style Light Bulbs, E26 Base 240LM Dimmable... \$
- ✓ Led Edison Bulb Dimmable, Brightown 6Pcs 60 Watt Equivalent E26 Base Vintage Led Filament Bulb 6W...

Network data analysis studies the model to gain insight and make predictions about these systems.

1. Evolution / changes

What new connections will form? (email auto-fill suggestions, rec. systems)

2. Clustering / partitioning / community detection

How to find groups of related nodes? (similar products, protein functions)

3. Spreading and traversing

How does stuff move over the network? (viruses or misinformation)

4. Ranking

Which things are important? (PageRank and its variants)

Real-world systems are composed of “higher-order” interactions that we often reduce to pairwise ones.



Communications

nodes are people/accounts
emails often have several recipients, not just one.



Physical proximity

nodes are people
people gather in groups



Commerce

nodes are products
Several products purchased at once



Drug compounds

nodes are substances
Drugs are composed of several substances

Frequently bought together



Total price: **\$55.96**

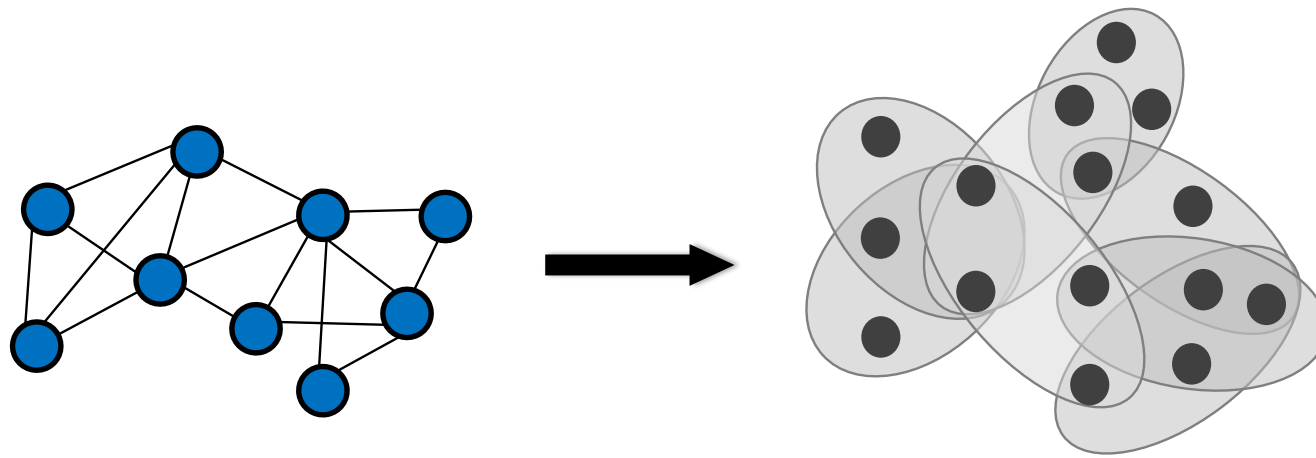
[Add all three to Cart](#)

[Add all three to List](#)

✓ **This item:** 6-Pack LED Dimmable Edison Light Bulbs 40W Equivalent Vintage Light Bulb, 2200K-2400K Wa

✓ Edison Light Bulbs, DOREShop 40Watt Antique Vintage Style Light Bulbs, E26 Base 240LM Dimmable... \$

✓ Led Edison Bulb Dimmable, Brightown 6Pcs 60 Watt Equivalent E26 Base Vintage Led Filament Bulb 6W...



What new insights does this give us?

We can ask the same network analysis questions while taking into account the higher-order structure.

1. Evolution / changes

What new connections will form? (email auto-fill suggestions, rec. systems)

2. Clustering / partitioning / community detection

How to find groups of related nodes? (similar products, protein functions)

3. Spreading and traversing

How does stuff move over the network? (viruses or misinformation)

4. Ranking

Which things are important? (PageRank and its variants)



w/ R. Abebe, M. Schaub,
J. Kleinberg, A. Jadbabaie

Higher-order Network Data Analysis

1. Temporal evolution of higher-order interactions.
Simplicial Closure and Higher-order Link Prediction, PNAS 2018.
2. Clustering in large networks of higher-order interactions.
Minimizing Localized Ratio Cuts in Hypergraphs, KDD, 2020.
3. Diffusions over higher-order interactions in networks.
Random walks on simplicial complexes and the normalized Hodge 1-Laplacian, SIAM Review, 2020.

We collected many datasets of timestamped simplices, where each simplex is a subset of nodes.

bit.ly/sc-holp-data

1. Coauthorship in different domains.
2. Emails with multiple recipients.
3. Tags on Q&A forums.
4. Threads on Q&A forums.
5. Contact/proximity measurements.
6. Musical artist collaboration.
7. Substance makeup and classification codes applied to drugs the FDA examines.
8. U.S. Congress committee memberships and bill sponsorship.
9. Combinations of drugs seen in patients in ER visits.

↑
4
↓
★

For a strongly regular graph, there are exactly 3 eigenvalues, all nonzero (I believe). One has multiplicity 1, which means the other two have pretty high multiplicities. There are tables that give these eigenvalues and multiplicities:

<http://www.win.tue.nl/~aeb/graphs/srg/srgtab1-50.html>

For example, the Schlaefli graph is order 27 but has an eigenvalue of order 20.

My question is, are there other known graphs (families, types, or just single graphs) that have large multiplicities of eigenvalues? When I check a random graph in Sage, it seems the max multiplicity is mostly 1.

(linear-algebra) (graph-theory) (eigenvalues-eigenvectors) (algebraic-graph-theory)

share cite edit

asked Nov 8 '11 at 13:31
Graphth
9,253 ● 2 ■ 28 ▲ 66

Seen this? Or this? — J. M. is not a mathematician Nov 8 '11 at 13:55

@J.M. Thanks, I will look at these. I'm not sure the second one applies. But, the first one seems to be a good one. — **Graphth** Nov 10 '11 at 21:26

add a comment

2 Answers active oldest votes

↑
4
↓
✓

+50

One class of examples are distance-regular graphs; strongly regular graphs are (essentially) distance-regular graphs with diameter. Distance-regular graphs can be constructed from Hadamard matrices, symmetric designs and linear codes.

If all eigenvalues of the adjacency matrix A of a graph are simple, then any matrix P that commutes with A must be a polynomial in A . It follows from this that all automorphisms have order dividing two, and also that the graph either is the complete graph K_2 or cannot be vertex transitive. So any vertex-transitive on more than two vertices has an eigenvalue which is not simple.

You can learn about these things in Biggs's "Algebraic Graph Theory", for example.

share cite edit

answered Nov 9 '11 at 0:48
Chris Godsil
10.8k ● 2 ■ 15 ▲ 34

<https://math.stackexchange.com/q/80181>

Thinking of higher-order data as a weighted projected graph with filled-in structures is a convenient viewpoint.

Data.

$t_1: \{1, 2, 3, 4\}$

$t_2: \{1, 3, 5\}$

$t_3: \{1, 6\}$

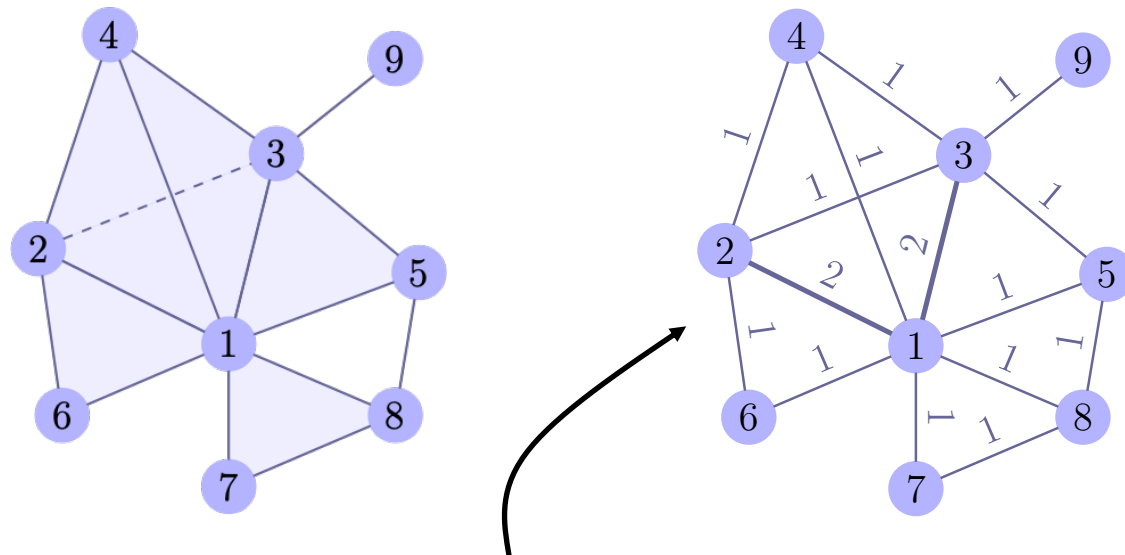
$t_4: \{2, 6\}$

$t_5: \{1, 7, 8\}$

$t_6: \{3, 9\}$

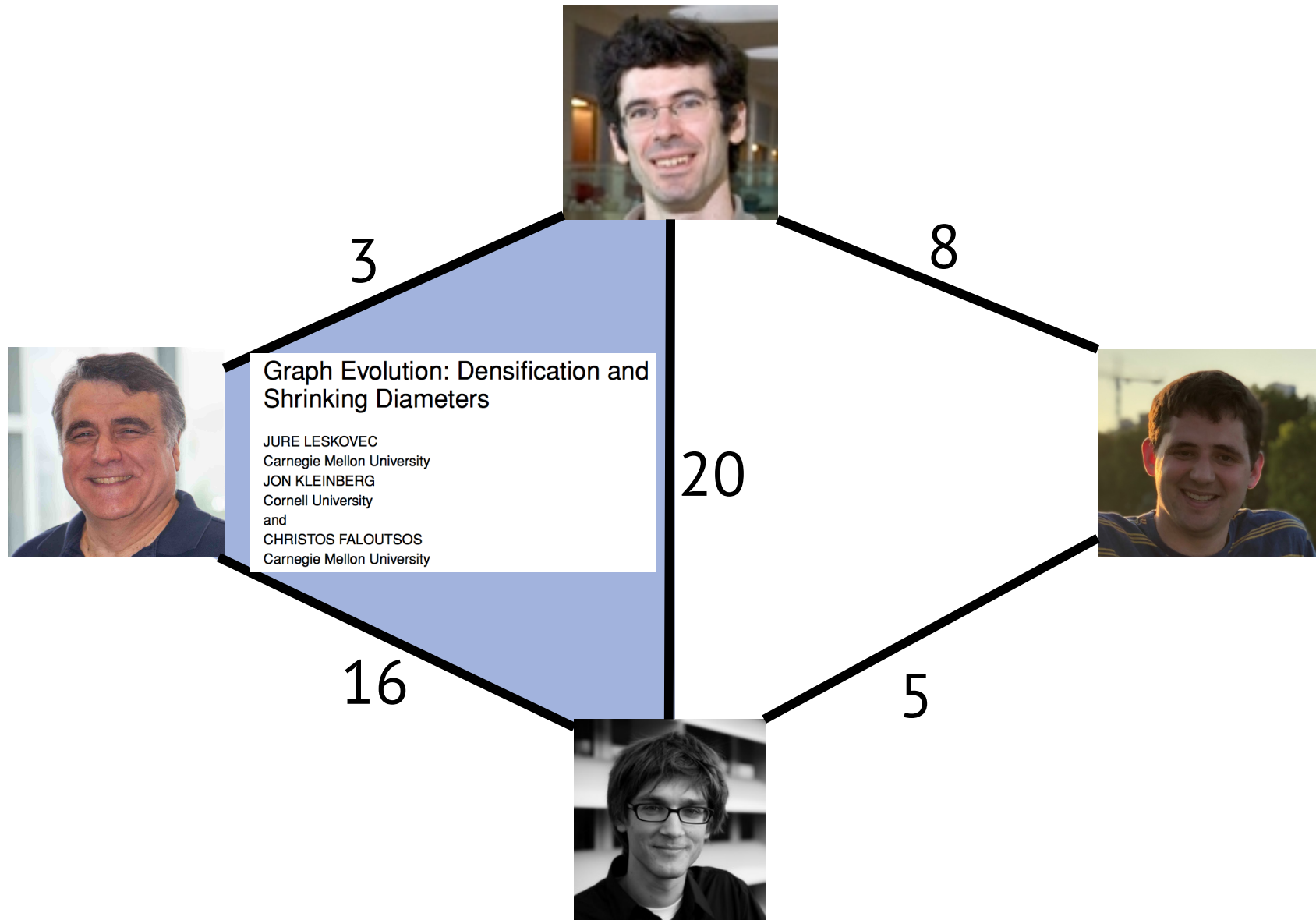
$t_7: \{5, 8\}$

$t_8: \{1, 2, 6\}$

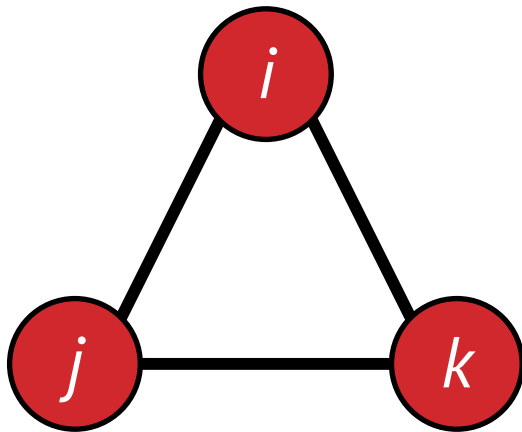


Projected graph \mathbf{W} .

$W_{ij} = \#$ of simplices containing nodes i and j .



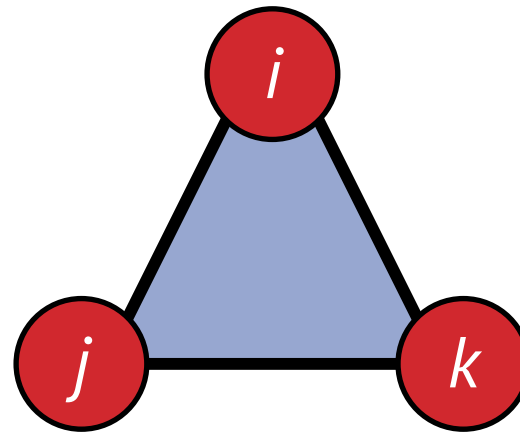
Warm-up. What's more common in data?



“Open triangle”

each pair has been in a simplex together but all 3 nodes have never been in the same simplex

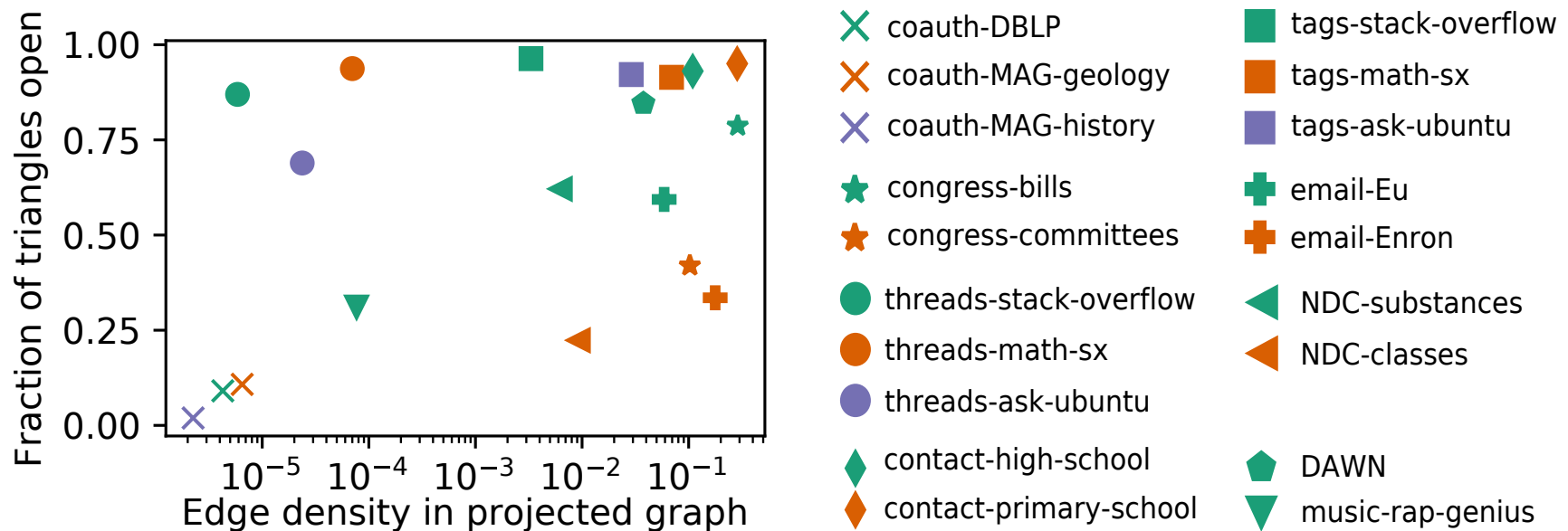
or



“Closed triangle”

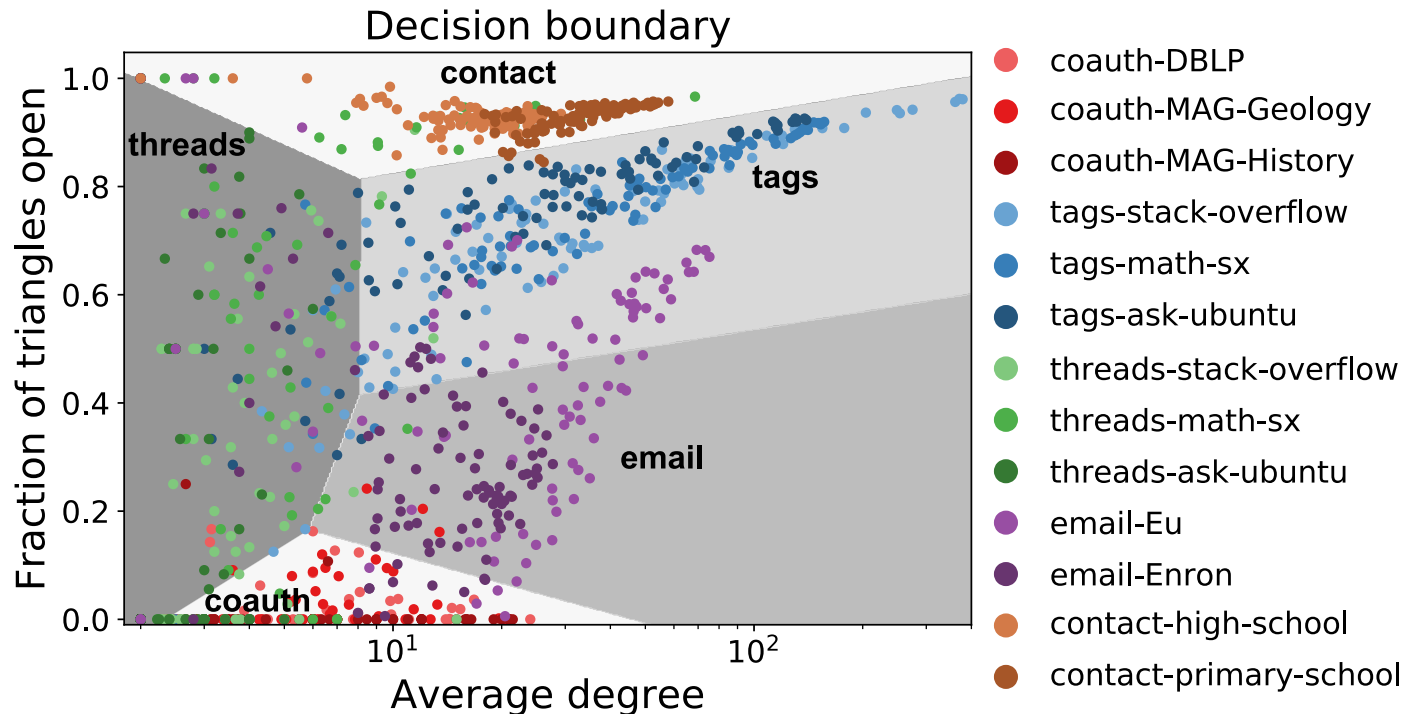
there is some simplex that contains all 3 nodes

There is lots of variation in the fraction of triangles that are open, but datasets from the same domain are similar.



See also Patania-Petri-Vaccarino (2017) for similar ideas in collaboration networks.

Dataset domain separation also occurs at the local level.



- Randomly sample 100 egonets per dataset and measure log of average degree and fraction of open triangles.
- Logistic regression model to predict domain (coauthorship, tags, threads, email, contact).
- 75% model accuracy vs. 21% with random guessing.

How do new simplices form?
Can we predict which simplices will form?

Groups of nodes go through trajectories until finally reaching a “simplicial closure.”

$t_1: \{1, 2, 3, 4\}$

$t_2: \{1, 3, 5\}$

$t_3: \{1, 6\}$

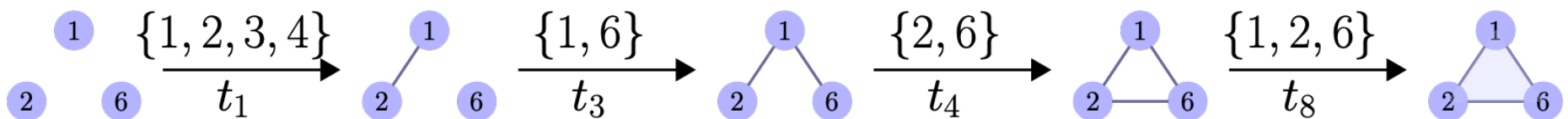
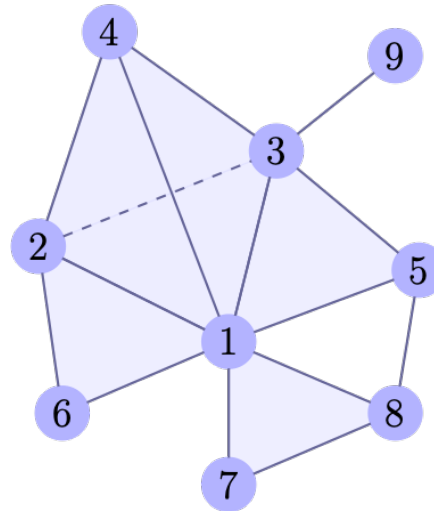
$t_4: \{2, 6\}$

$t_5: \{1, 7, 8\}$

$t_6: \{3, 9\}$

$t_7: \{5, 8\}$

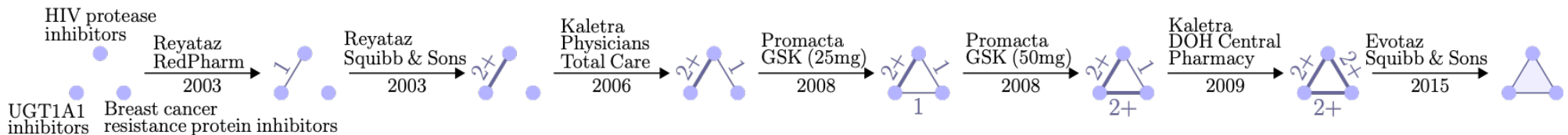
$t_8: \{1, 2, 6\}$



For this talk, we will focus on simplicial closure on 3 nodes.

Groups of nodes go through trajectories until finally reaching a “simplicial closure event.”

Substances in marketed drugs recorded in the National Drug Code directory.

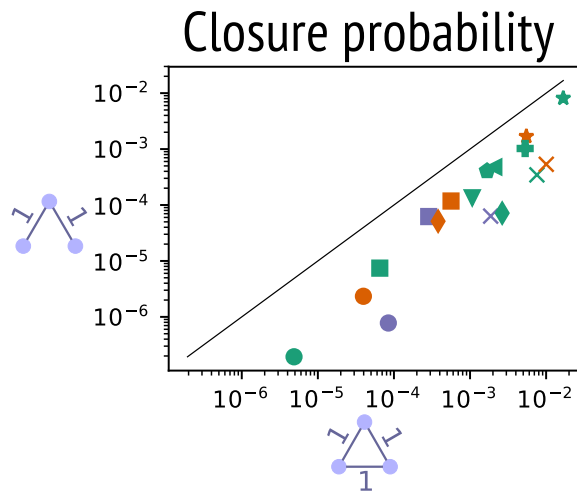


We bin weighted edges into “weak” and “strong ties” in the projected graph W .
 W_{ij} = # of simplices containing nodes i and j .

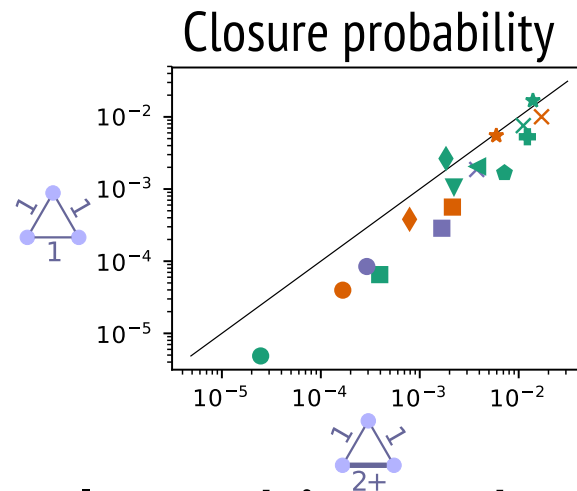
- **Weak ties.** $W_{ij} = 1$ (one simplex contains i and j)
- **Strong ties.** $W_{ij} \geq 2$ (at least two simplices contain i and j)

Simplicial closure depends on structure in projected graph.

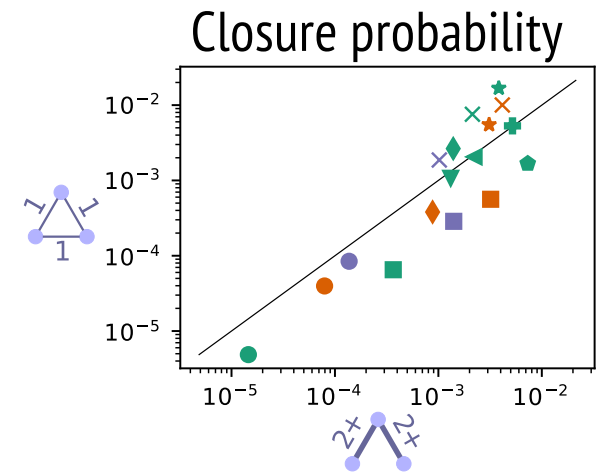
- First 80% of the data (in time) \rightarrow record configurations of triplets not in closed triangle.
- Remainder of data \rightarrow find fraction that are now closed triangles.



**Increased edge density
increases closure probability.**



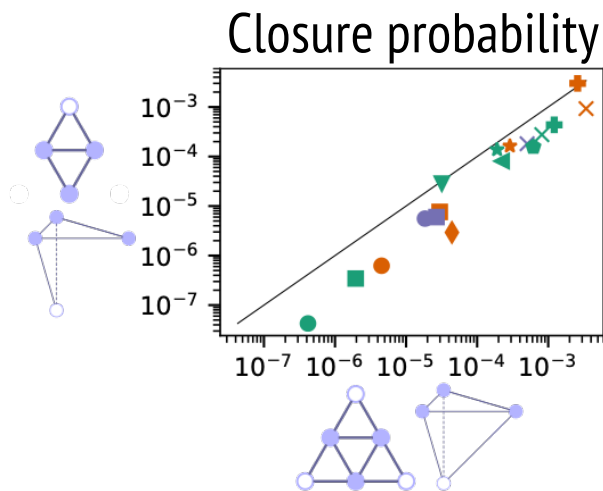
**Increased tie strength
increases closure probability.**



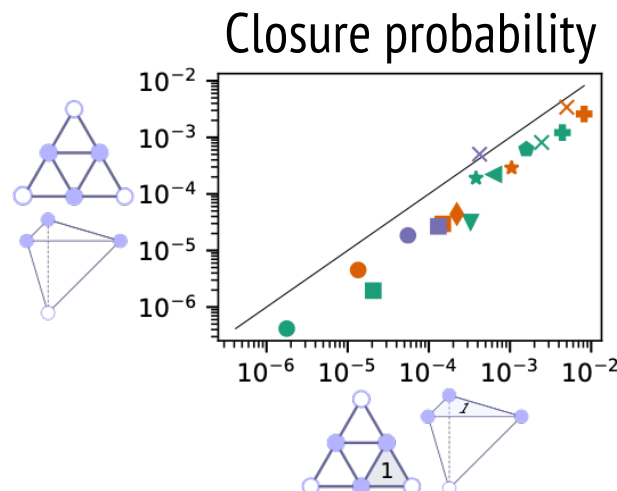
**Tension between edge
density and tie strength.**

Left and middle observations are consistent with theory and empirical studies of *social* networks.
[Granovetter 73; Kossinets-Watts 06; Backstrom+ 06; Leskovec+ 08]

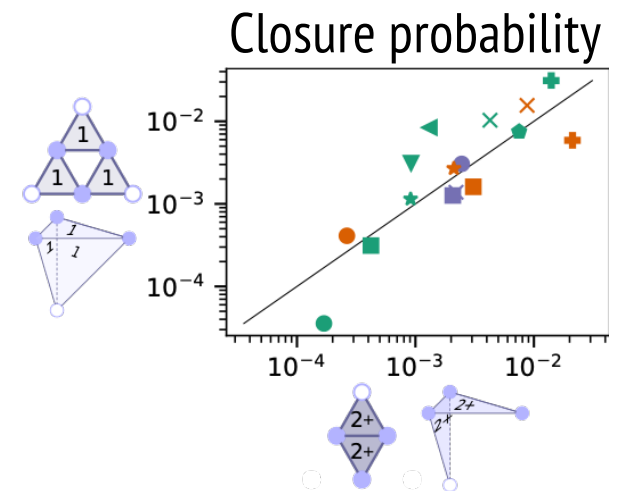
Simplicial closure on 4 nodes is similar to on 3 nodes, just “up one dimension.”



Increased edge density increases closure probability.



Increased **simplicial tie strength** increases closure probability.



Tension b/w edge density simplicial tie strength.

We proposed “higher-order link prediction” as a framework to evaluate models for closure.

Data.

$t_1: \{1, 2, 3, 4\}$

$t_2: \{1, 3, 5\}$

$t_3: \{1, 6\}$

$t_4: \{2, 6\}$

$t_5: \{1, 7, 8\}$

$t_6: \{3, 9\}$

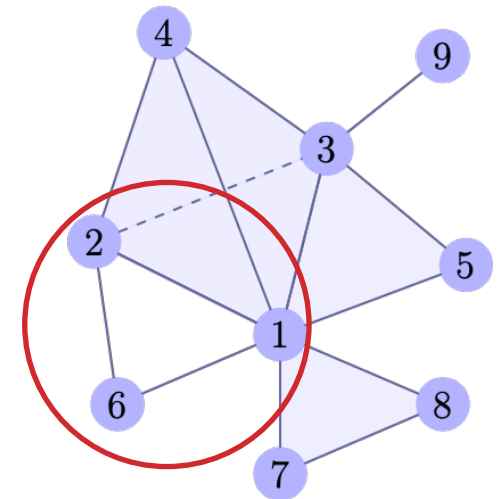
$t_7: \{5, 8\}$

$t_8: \{1, 2, 6\}$

t

- Observe simplices up to time t .
- Predict which groups of > 2 nodes will appear after time t .

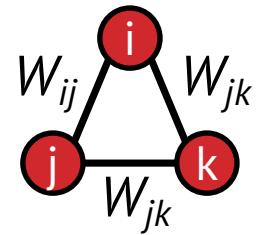
We predict structure that graph models would not even consider!



Our structural analysis tells us what we should be looking at for prediction.

1. Edge density matters!

→ focus our attention on predicting which open triangles become closed triangles
(intelligently reduce search space.)



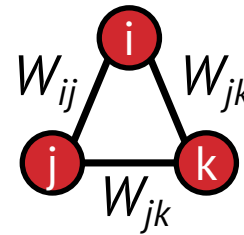
2. Tie strength matters!

→ various ways of incorporating this information

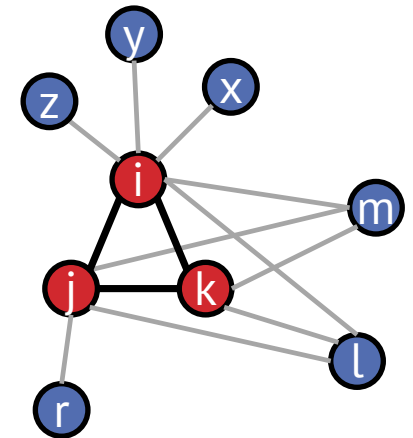
For every open triangle, we assign a score function on first 80% of data based on structural properties.

Score $s(i, j, k)$...

1. is a function of W_{ij}, W_{jk}, W_{ik}
arithmetic mean, harmonic mean, etc.
2. looks at common neighbors of the three nodes.
generalized Jaccard, Adamic-Adar, etc.
3. uses “whole-network” similarity scores on projected graph
sum of PageRank or Katz scores amongst edges
4. is learned from data
logistic regression model with features



$$\text{score}_p(i, j, k) = (W_{ij}^p + W_{jk}^p + W_{ik}^p)^{1/p}$$



$$N(i) = \{j, k, l, m, x, y, z\}$$

$$N(j) = \{i, k, l, m, r\}$$

$$N(k) = \{i, j, l, m\}$$

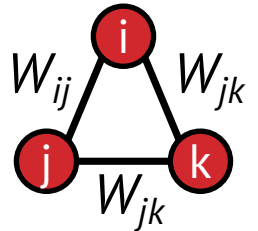
After computing scores, predict that open triangles with highest scores will be closed triangles in final 20% of data.

Table 2: Open triangle closure prediction performance based on several score functions: random (Rand.); harmonic, geometric, and arithmetic means of the 3 edge weights (Eqs. (19) to (21)); 3-way common neighbors (Common, Eq. (22)); 3-way Jaccard coefficient (Jaccard, Eq. (23)); 3-way Adamic-Adar (A-A, Eq. (24)); projected graph degree and simplicial degree preferential attachment (PGD-PA, Eq. (25) and SD-PA, Eq. (25)); unweighted and weighted Katz similarity (Katz, Eq. (29) and W-Katz, Eq. (30)); unweighted and weighted personalized PageRank (U-PPR, Eq. (34) and W-PPR, Eq. (35)); simplicial personalized PageRank (S-PPR, Eq. (42)); the two missing entries are cases where computations did not finish within 2 weeks); and a feature-based supervised method logistic regression (Log. reg.). Performance is AUC-PR relative to the random baseline. The random baseline is listed in absolute terms and equals the fraction of open triangles that close.

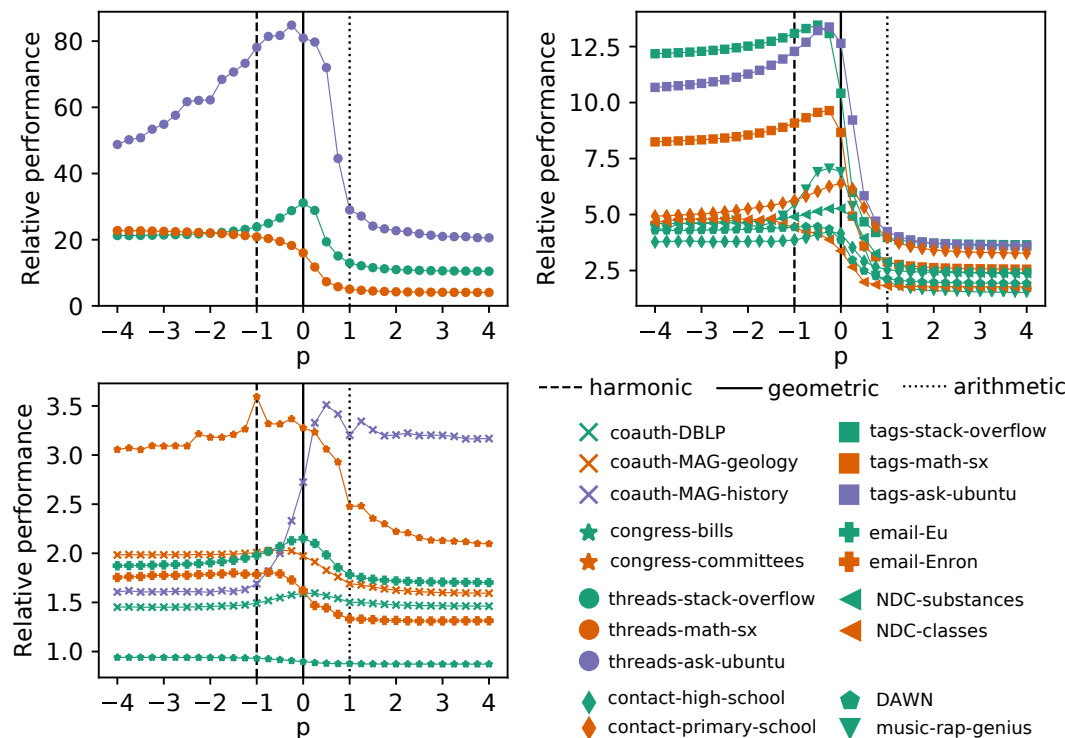
Dataset	Rand.	Harm. mean	Geom. mean	Arith. mean	Common	Jaccard	A-A	PGD-PA	SD-PA	U-Katz	W-Katz	U-PPR	W-PPR	S-PPR	Log. reg.
coauth-DBLP	1.68e-03	1.49	1.59	1.50	1.33	1.84	1.60	0.74	0.74	0.97	1.51	1.62	1.83	1.21	3.37
coauth-MAG-History	7.16e-04	1.69	2.72	3.20	5.11	2.24	5.82	1.50	2.49	6.30	3.40	1.66	1.88	1.35	6.75
coauth-MAG-Geology	3.35e-03	2.01	1.97	1.69	2.43	1.84	2.71	1.31	0.97	1.99	1.74	1.06	1.26	0.94	4.74
music-rap-genius	6.82e-04	5.44	6.92	1.98	1.85	1.62	2.10	1.82	2.15	1.93	2.00	1.78	2.09	1.39	2.67
tags-stack-overflow	1.84e-04	13.08	10.42	3.97	6.45	9.43	6.63	3.37	2.74	2.95	3.60	1.08	1.85	–	3.37
tags-math-sx	1.08e-03	9.08	8.67	2.88	6.19	9.37	6.34	3.48	2.81	4.53	2.71	1.19	1.55	1.86	13.99
tags-ask-ubuntu	1.08e-03	12.29	12.64	4.24	7.15	4.96	7.51	7.48	5.63	7.10	4.15	1.75	2.54	1.19	7.48
threads-stack-overflow	1.14e-05	23.85	31.12	12.97	2.73	3.85	3.19	5.20	3.89	1.06	11.54	1.66	4.06	–	1.53
threads-math-sx	5.63e-05	20.86	16.01	5.03	25.08	28.13	23.32	10.46	7.46	11.04	4.86	0.90	1.18	0.61	47.18
threads-ask-ubuntu	1.31e-04	78.12	80.94	29.00	21.04	2.80	30.82	7.09	6.62	16.63	32.31	0.94	1.51	1.78	9.82
NDC-substances	1.17e-03	4.90	5.27	2.90	5.92	3.36	5.97	4.76	4.46	5.35	2.93	1.39	1.83	1.86	8.17
NDC-classes	6.72e-03	4.43	3.38	1.82	1.27	1.19	0.99	0.94	2.14	0.92	1.34	0.78	0.91	2.45	0.62
DAWN	8.47e-03	4.43	3.86	2.13	4.73	3.76	4.77	3.76	1.45	4.61	2.04	1.57	1.37	1.55	2.86
congress-committees	6.99e-04	3.59	3.28	2.48	4.83	2.49	5.04	1.06	1.31	3.21	2.59	1.50	3.89	2.13	7.67
congress-bills	1.71e-04	0.93	0.90	0.88	0.65	1.23	0.66	0.60	0.55	0.60	0.78	3.16	1.07	6.01	107.19
email-Enron	1.40e-02	1.78	1.62	1.33	0.85	0.83	0.87	1.27	0.83	0.99	1.28	3.69	3.16	2.02	0.72
email-Eu	5.34e-03	1.98	2.15	1.78	1.28	2.69	1.37	0.88	1.55	1.01	1.79	1.59	1.75	1.26	3.47
contact-high-school	2.47e-03	3.86	4.16	2.54	1.92	3.61	2.00	0.96	1.13	1.72	2.53	1.39	2.41	0.78	2.86
contact-primary-school	2.59e-03	5.63	6.40	3.96	2.98	2.95	3.21	0.92	0.94	1.63	4.02	1.41	4.31	0.93	6.91

A few lessons learned from applying these ideas.

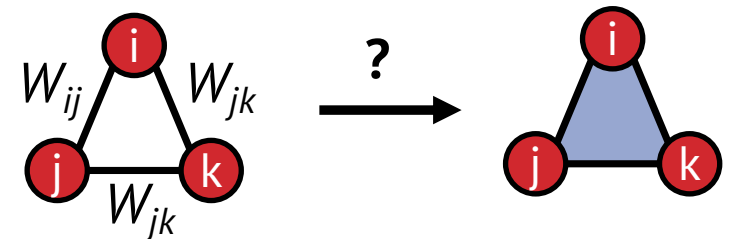
1. We can predict pretty well on *all* datasets using *some* simple method.
→ 4x to 107x better than random w/r/t mean average precision
depending on the dataset/method
(only predicting on open triangles)
2. Thread co-participation and co-tagging on stack exchange are consistently easy to predict with the harmonic mean.
3. Simple averaging W_{ij} , W_{jk} , and W_{ik} consistently performs well.



Generalized means of edges weights are often good predictors of new 3-node simplices appearing.



$$\text{score}_p(i, j, k) = (W_{ij}^p + W_{jk}^p + W_{ik}^p)^{1/p}$$



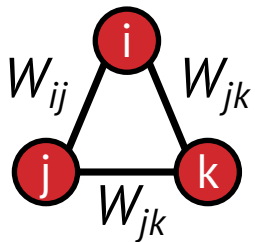
Good performance from this local information is a deviation from classical link prediction, where methods that use long paths (e.g., PageRank) perform well [Liben-Nowell & Kleinberg 07]. For structures on k nodes, the subsets of size $k-1$ contain rich information only when $k > 2$.

If we only need the top-k weighted triangles, we have fast algorithms for finding them.



w/ R. Kumar, P. Liu, M. Charikar

$$\text{score}_p(i, j, k) \\ = (W_{ij}^p + W_{jk}^p + W_{ik}^p)^{1/p}$$

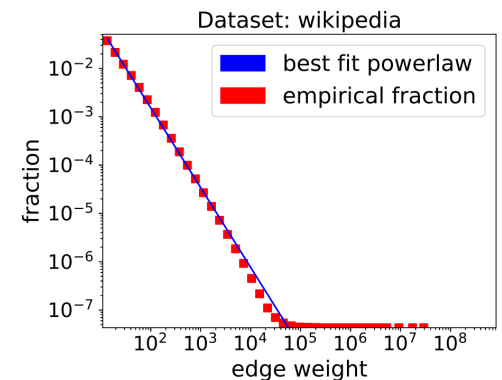


Simple (incorrect) algorithm.

1. Throw out edges with weight $< t$.
2. Find triangles in remainder.

Better (correct) algorithm.

1. Dynamically choose threshold.
2. Careful pruning.

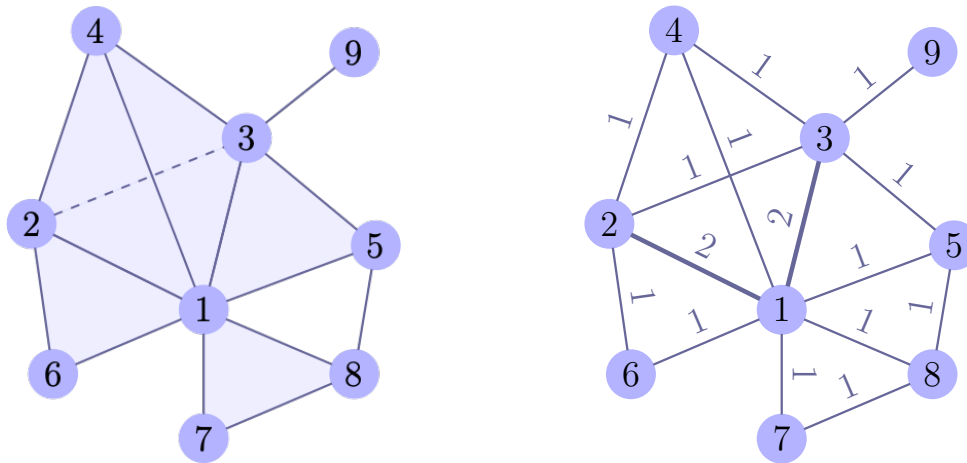


We often only need the top-k weighted triangles, and we have fast algorithms for finding them.

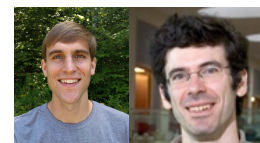
dataset	# nodes	# edges	Fast enumeration	Fast top- k ($k = 1000$)
			(running time in seconds)	
Spotify co-listens	3.6M	1.93B	too long	30
MAG co-authorship	173M	544M	596	16
AMINER co-authorship	93M	324M	255	10
Ethereum transactions	38M	103M	91	33

Higher-order data is pervasive!

1. There are commonalities in temporal evolution. Generative models?
2. There is lots of signal in subsets! Unique to higher-order...
3. Please develop neural embeddings to out-perform our baselines. 😊



- Simplicial Closure and Higher-order Link Prediction. Austin R. Benson, Rediet Abebe, Michael T. Schaub, Ali Jadbabaie, and Jon Kleinberg. Proc. Natl. Acad. Sci. U.S.A., 2018. 🐱 github.com/arbenson/ScHoLP-Tutorial
- Retrieving Top Weighted Triangles in Graphs. Raunak Kumar, Paul Liu, Moses Charikar, and Austin R. Benson. Proc. Of WSDM, 2020. 🐱 github.com/raunakkmr/Retrieving-top-weighted-triangles-in-graphs

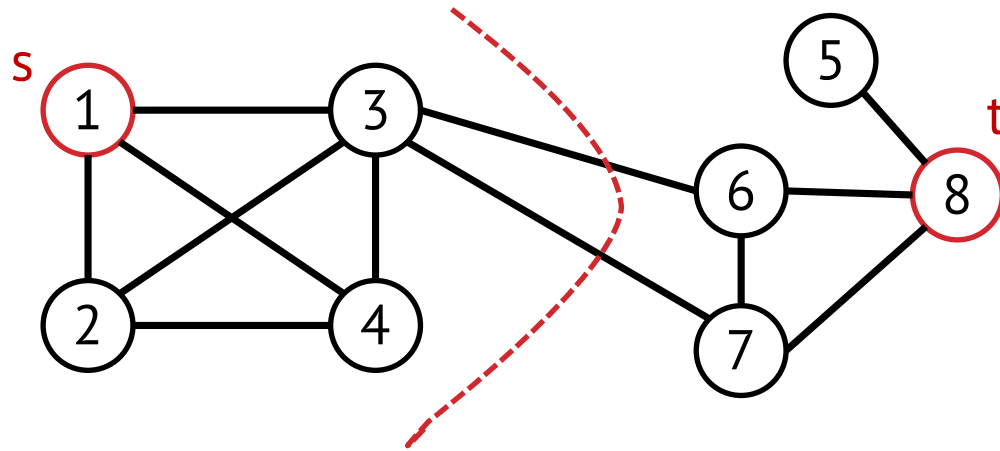


w/ Nate Veldt, J. Kleinberg

Higher-order Network Data Analysis

1. Temporal evolution of higher-order interactions.
Simplicial Closure and Higher-order Link Prediction, PNAS 2018.
2. Clustering in large networks of higher-order interactions.
Minimizing Localized Ratio Cuts in Hypergraphs, KDD, 2020.
3. Diffusions over higher-order interactions in networks.
Random walks on simplicial complexes and the normalized Hodge 1-Laplacian, SIAM Review, 2020.

Graph minimum s-t cuts are fundamental.



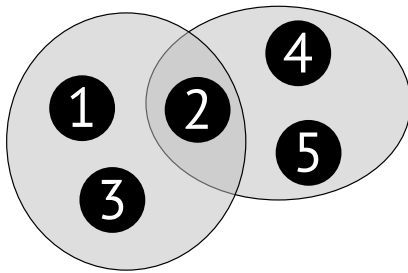
minimize $\sum_{s \in S, t \in T} c_{st}$ cut(S)
subject to $s \in S, t \notin S$.

poly-time algorithms!

- Maximum flow / min s-t cut [Ford, Fulkerson, Dantzig 1950s]
- Densest subgraph [Goldberg 84; Shang+ 18]
- Graph-based semi-supervised learning algorithms [Blum-Chawla 01]
- Local graph clustering [Andersen-Lang 08; Orecchia-Zhu 14; Veldt+ 16]

Real-world systems are composed of “higher-order” interactions that we can model with hypergraphs.

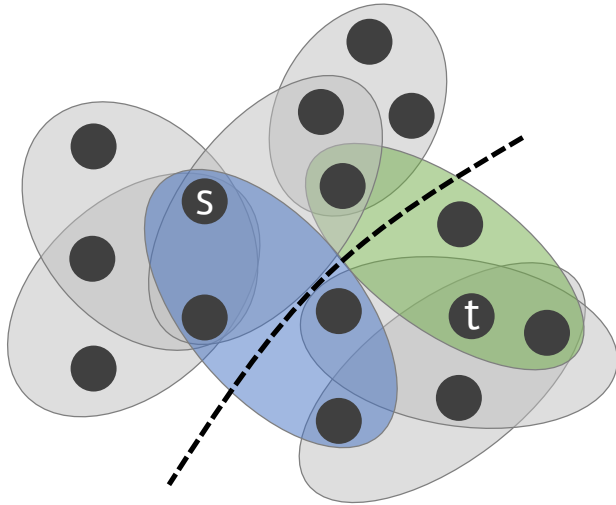
$\mathcal{H} = (V, E)$, edge $e \in E$ is a subset of V ($e \subset V$)



$$V = \{1, 2, 3, 4, 5\}$$

$$E = \{\{1, 2, 3\}, \{2, 4, 5\}\}$$

What is a hypergraph minimum s-t cut?

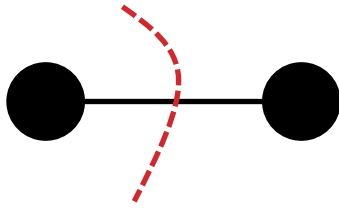


Should we treat the $2/2$ split differently from the $1/3$ split?

Historically, no. [Lawler 73, Ihler+ 93]

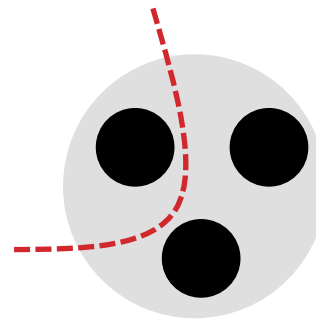
More recently, yes.

[Li-Milenkovic 17, Veldt-Benson-Kleinberg 20]



edge in a graph

Must be split $1/1$.

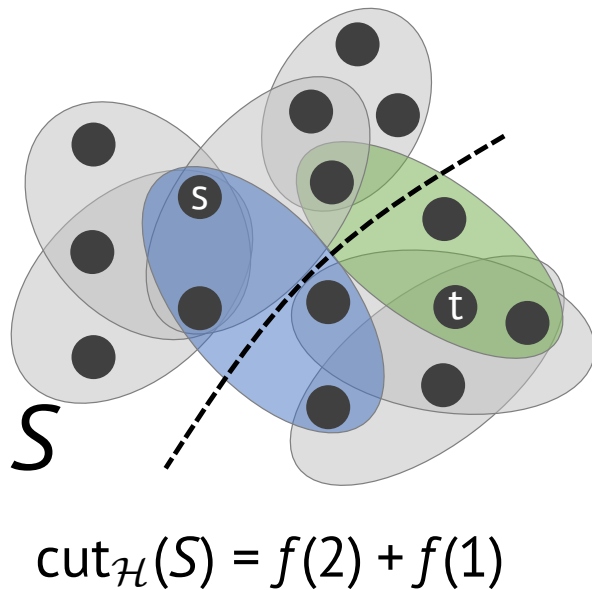


size-3 hyperedges

“Only one way to split a triangle”

[Benson+ 16; Li-Milenkovic 17; Yin+ 17]

We model hypergraph cuts with splitting functions.



Given a cut defined by S ,
we incur penalty of $\mathbf{w}_e(e \cap S)$
at each hyperedge e .

Hypergraph minimum s-t cut problem.

minimize $S \subset V$ $\sum_{e \in E} \mathbf{w}_e(e \cap S) \equiv \text{cut}_{\mathcal{H}}(S)$
subject to $s \in S, t \notin S$.

Cardinality-based splitting functions.

Non-negativity $\mathbf{w}_e(A) \geq 0$.

Non-split ignoring $\mathbf{w}_e(e) = \mathbf{w}_e(\emptyset) = 0$.

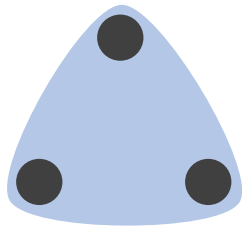
C-B $\mathbf{w}_e(A) = f(\min(|A|, |A \setminus e|))$.

Cardinality-based splitting functions appear throughout the literature.

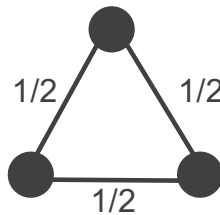
All-or-nothing	$\mathbf{w}_e(A) = \begin{cases} 0 & \text{if } A \in \{e, \emptyset\} \\ 1 & \text{otherwise} \end{cases}$	[Lawler 73; Ihler+ 93; Yin+ 17]
Linear penalty	$\mathbf{w}_e(A) = \min\{ A , e \setminus A \}$	[Hu-Moerder 85; Heuer+ 18]
Quadratic penalty	$\mathbf{w}_e(A) = A \cdot e \setminus A $	[Agarwal+ 06; Zhou+ 06; Benson+ 16]
Discount cut	$\mathbf{w}_e(A) = \min\{ A ^\alpha, e \setminus A ^\alpha\}$	[Yaros- Imielinski 13]
L-M submodular	$\mathbf{w}_e(A) = \frac{1}{2} + \frac{1}{2} \cdot \min \left\{ 1, \frac{ A }{\lfloor \alpha e \rfloor}, \frac{ e \setminus A }{\lfloor \alpha e \rfloor} \right\}$	[Li-Milenkovic 18]

We solve hypergraph cut problems with graph reductions.

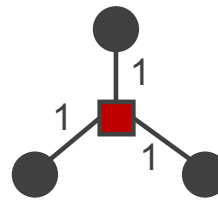
Gadgets (expansions) model a hyperedge with a small graph.



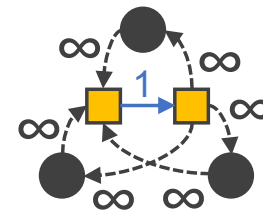
hyperedge



clique expansion

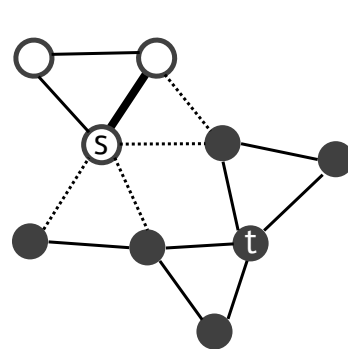
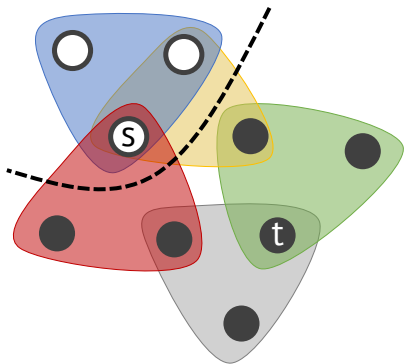


star expansion

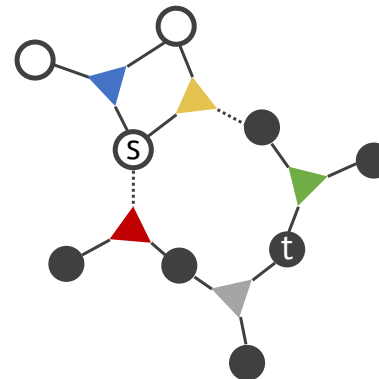


Lawler gadget [1973]

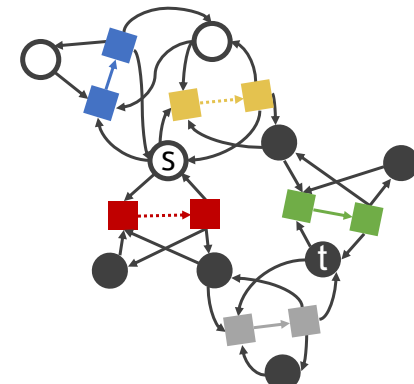
In a graph reduction, we first replace all hyperedges with graph gadgets...



Quadratic penalty
 $f(i) = i (|e| - i)$



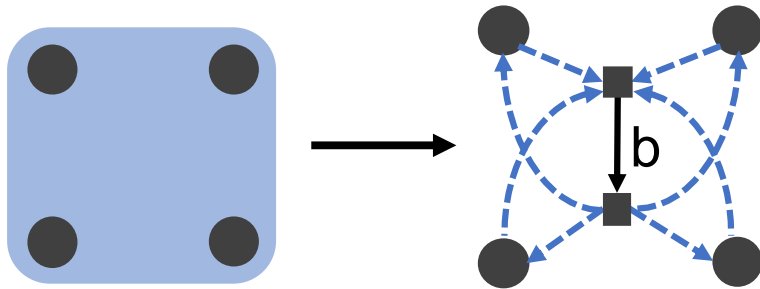
Linear penalty
 $f(i) = i$



All-or-nothing
 $f(0) = 0, \text{ o/w } f(i) = 1$

... then solve the (min s-t cut) problem *exactly* on the graph,
and finally convert the solution to a hypergraph solution.

We made a new gadget for C-B splitting functions.



C-B $w_e(A) = f(\min(|A|, |e \setminus A|))$.
This gadget models $\min(|A|, |e \setminus A|, b)$.

Theorem [Veldt-Benson-Kleinberg 20a]. Nonnegative linear combinations of the C-B gadget can model any submodular cardinality-based splitting function.
(F is submodular on X if $F(A \cap B) + F(A \cup B) \leq F(A) + F(B)$ for any $A, B \subseteq X$.)

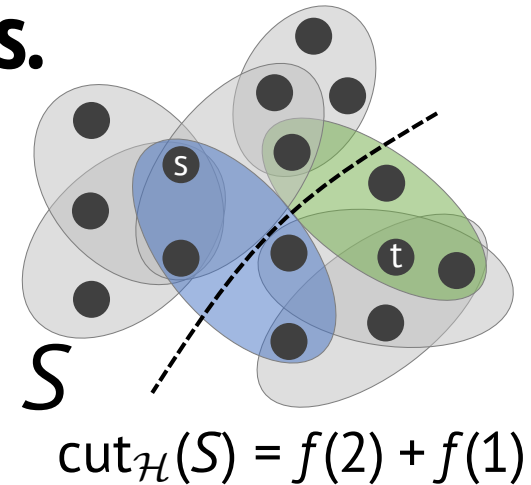
Submodularity is key to efficient algorithms.

Cardinality-based splitting functions.

Non-negativity $\mathbf{w}_e(A) \geq 0$.

Non-split ignoring $\mathbf{w}_e(e) = \mathbf{w}_e(\emptyset) = 0$.

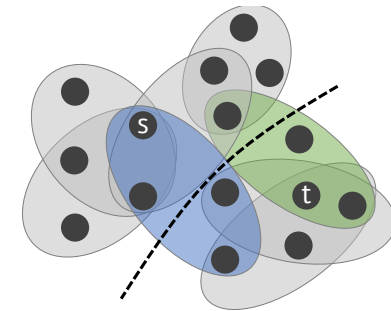
C-B $\mathbf{w}_e(A) = f(\min(|A|, |A \setminus e|))$.



Theorem [Veldt-Benson-Kleinberg 20a]. The hypergraph min s-t cut problem with a cardinality-based splitting function is graph-reducible (via gadgets) *if and only if* the splitting function is submodular.

What happens when the splitting function isn't submodular?
Can we use some other algorithm?

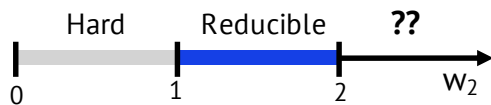
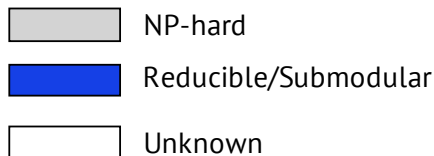
Unlike graph min s-t cut, hypergraph min s-t cut can be NP-hard.



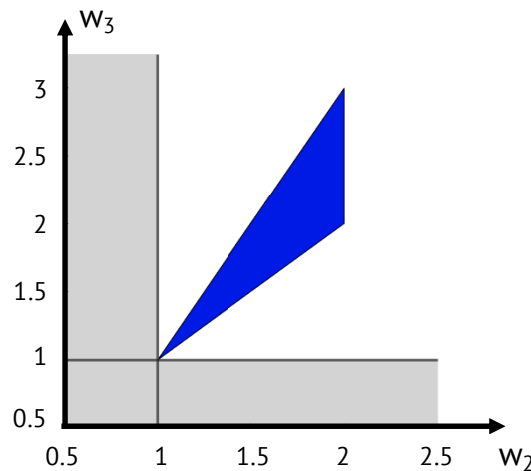
$$\text{cut}_{\mathcal{H}}(S) = f(2) + f(1) = w_2 + 1$$

Theorem [Veldt-Benson-Kleinberg 20a]. For C-B splitting functions,

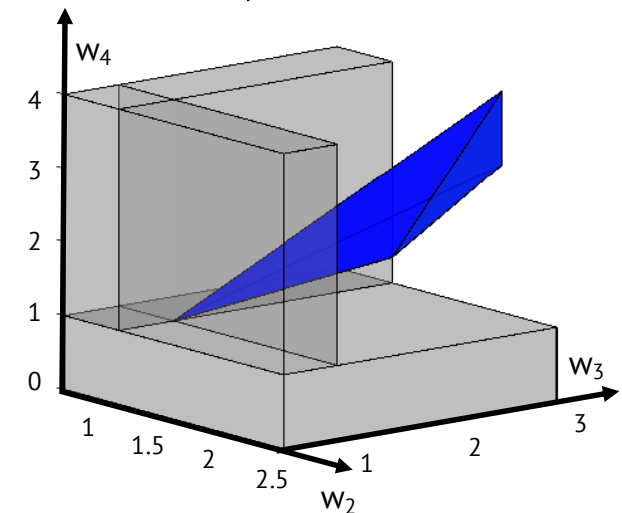
$$w_1 = 1$$



max hyperedge size 4 or 5



max hyperedge size 6 or 7



max hyperedge size 8 or 9

Open Question: For 4-uniform hypergraphs, is there an efficient algorithm to find the minimum s-t cut with no 2-2 splits ($w_1 = 1, w_2 = \infty$).

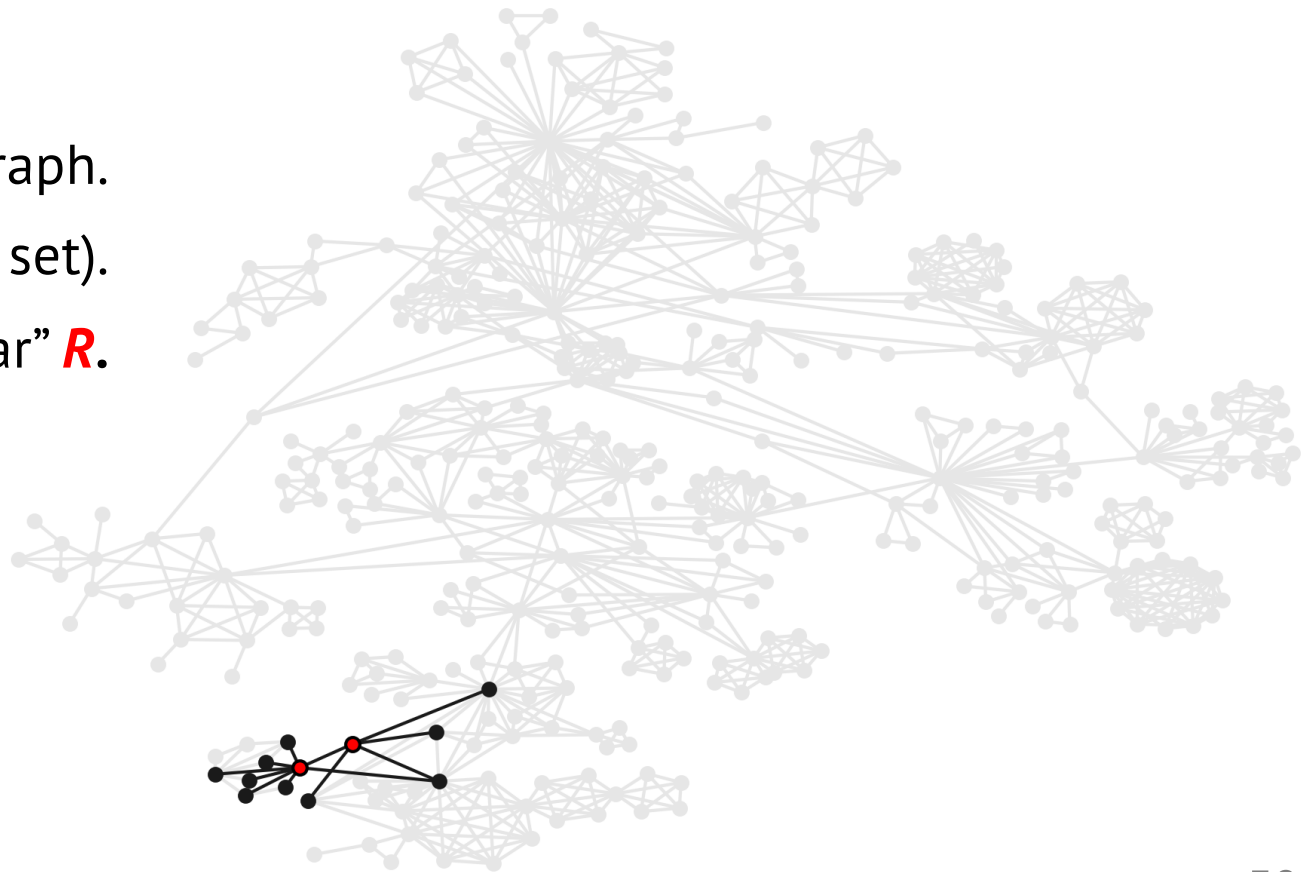
How can we use this framework to enable new data science algorithms?

Background. Local clustering has been studied extensively in graphs, but not much in hypergraphs.

$G = (V, E)$ is a graph.

$R \subseteq V$ (**R**eference or **s**eed set).

Finds a “good” cluster S “near” R .



Background. Flow-based methods minimize a localized variant of conductance.

minimize node sets S

$$\phi_R(S) = \frac{\text{cut}(S)}{\text{vol}(S \cap R) - \epsilon \text{vol}(S \cap \bar{R})}$$

$\text{vol}(T)$ = sum of degrees in T . Rewards high overlap with R .

Rewards contained clusters

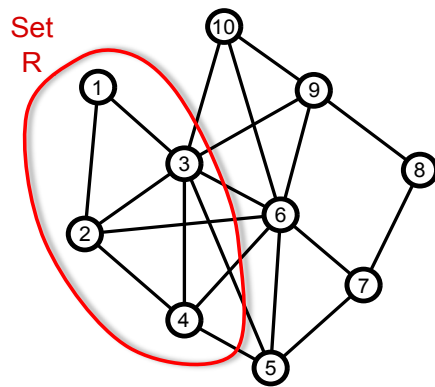
Penalizes nodes outside R .

FAST ALGORITHMS FOR EXACT MINIMIZATION!

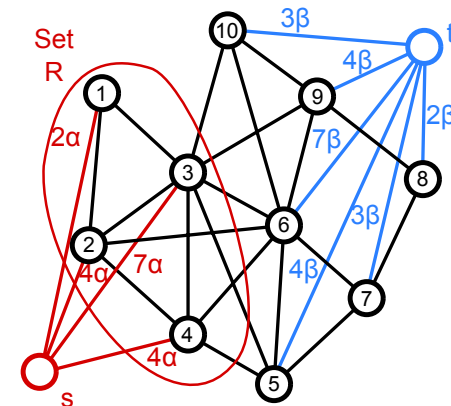
Max Flow. Quot. Imp. (Lang, Rao, 2004)
Flow-Improve (Andersen, Lang 2008)
Local-Improve (Orecchia, Allen-Zhou 2014)
SimpleLocal (Veldt, Gleich, Mahoney 2016)
FlowSeed (Veldt, Klymko, Gleich 2019)
Great survey paper! (Fountoulakis et al. 2020)

Background. Flow methods repeatedly solve min-cut problems on an auxiliary graph. [Andersen-Lang 08, Orecchia-Zhou 14, Veldt+ 16]

Is $\phi_R(S) < \alpha$ for any S ?



Construct G'



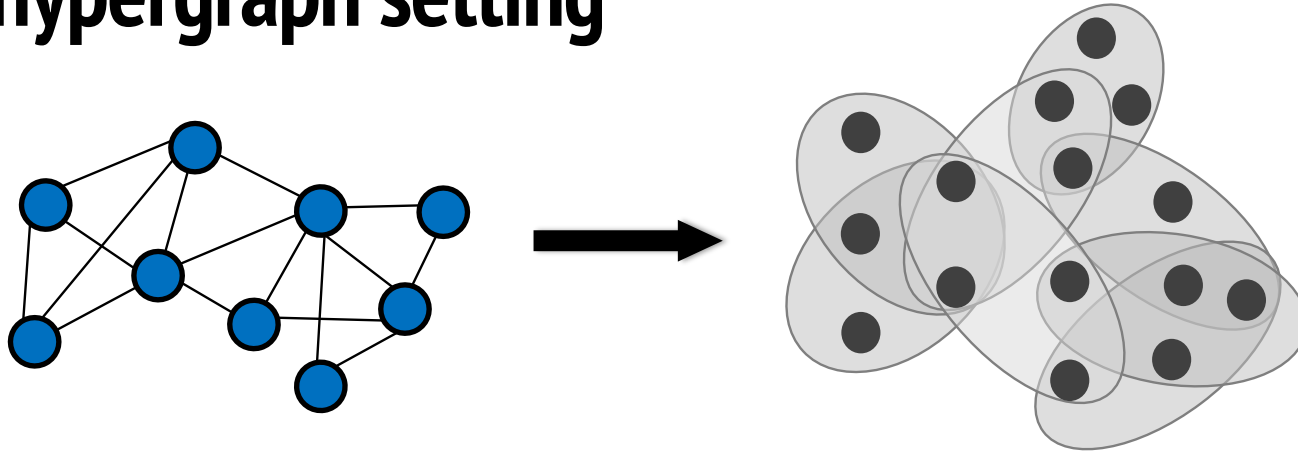
Connect **R** to a source node **(s)**; edges weighted with respect to α .

Connect **$V \setminus R$** to a sink node **(t)**; edges weighted with respect to $\beta = \alpha\epsilon$.

Compute min s-t cut of G' .

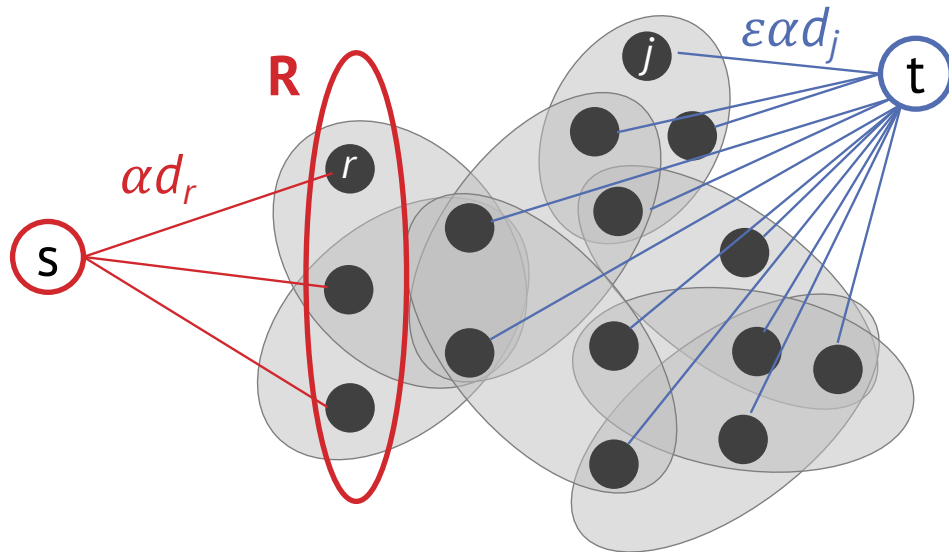
$$\phi_R(S) < \alpha \iff \text{min s-t cut of } G' < \alpha \text{vol}(R)$$

We generalized local flow-based techniques to the hypergraph setting



- We introduce **localized hypergraph conductance**
- We can minimize it *exactly* with our hypergraph min s-t cuts framework
- Strongly-local runtime! (Only depends on size of seed set)
- *Normalized cut* improvement guarantees \longrightarrow *The analysis provides even new guarantees for the graph case!*

We define hypergraph s-t cut problems similar to the ones used in the graph case.



Hypergraph cut function

$$\text{HLC}_{R,\epsilon}(S) = \frac{\text{cut}_{\mathcal{H}}(S)}{\text{vol}_{\mathcal{H}}(S \cap R) + \epsilon \text{vol}_{\mathcal{H}}(S \cap \bar{R})}$$

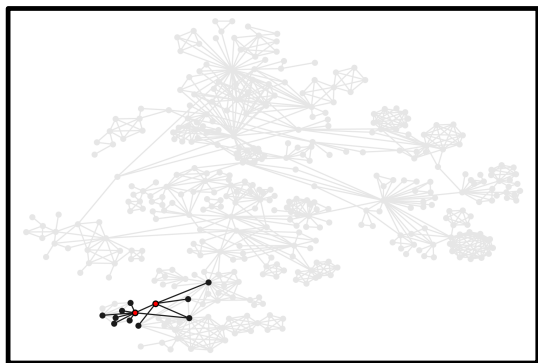
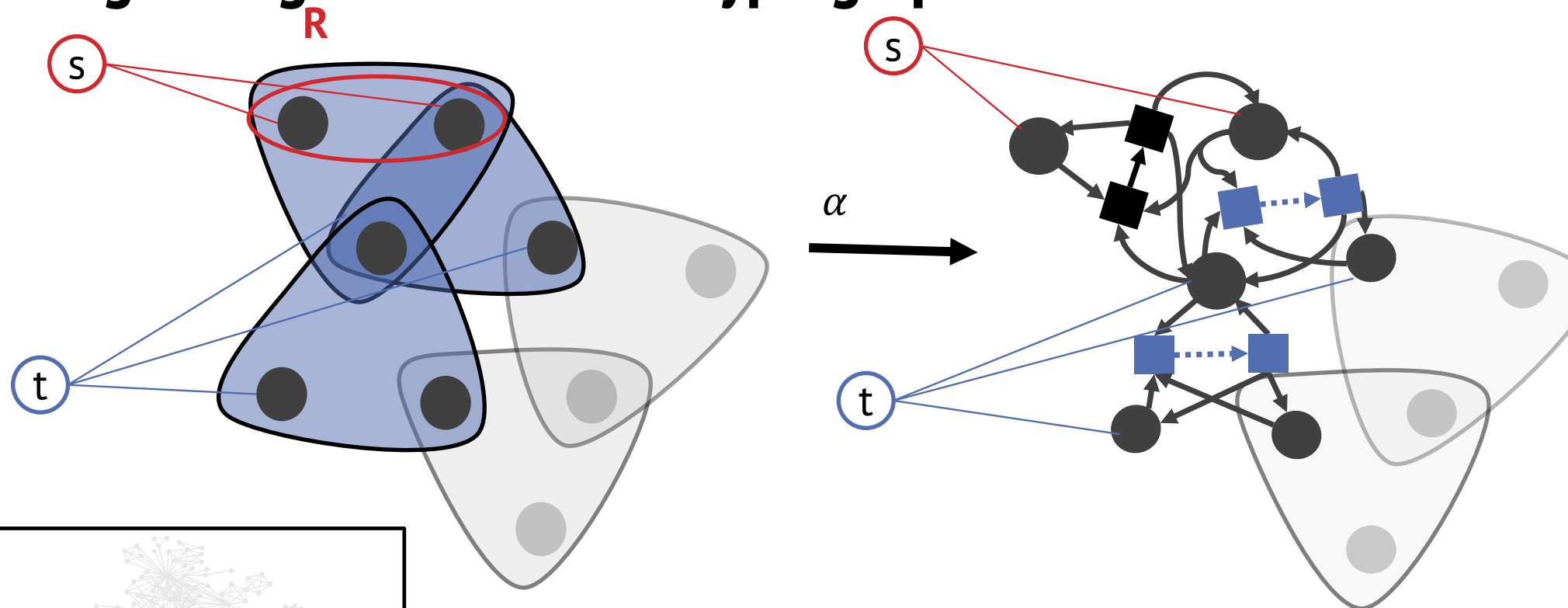
Encourage overlap with reference set. Discourage overlap outside reference set

$d_i = \#$ hyperedges node r is in

$$\rightarrow \text{vol}_{\mathcal{H}}(S) = \sum_{i \in S} d_i$$

Theorem [Veldt-Benson-Kleinberg 20b]. We can repeatedly solve min hypergraph s-t cut problems with different α to **exactly minimize** the hypergraph localized conductance (HLC) exactly.

We *carefully* apply graph reduction techniques to growing subsets of the hypergraph.



Theorem [Veldt-Benson-Kleinberg 20b]. Strong locality.

Can make this algorithm run in time proportional to the size of seed set (does not look at the full hypergraph).

We prove new normalized cut guarantees that are new even for the graph case.

Normalized cut is another ratio-cut objective related to conductance.

$$\phi(S) = \frac{\text{cut}(S)}{\text{vol}(S)} + \frac{\text{cut}(S)}{\text{vol}(\bar{S})}$$

Theorem [Veldt-Benson-Kleinberg 20b]. *Normalized cut improvement.*

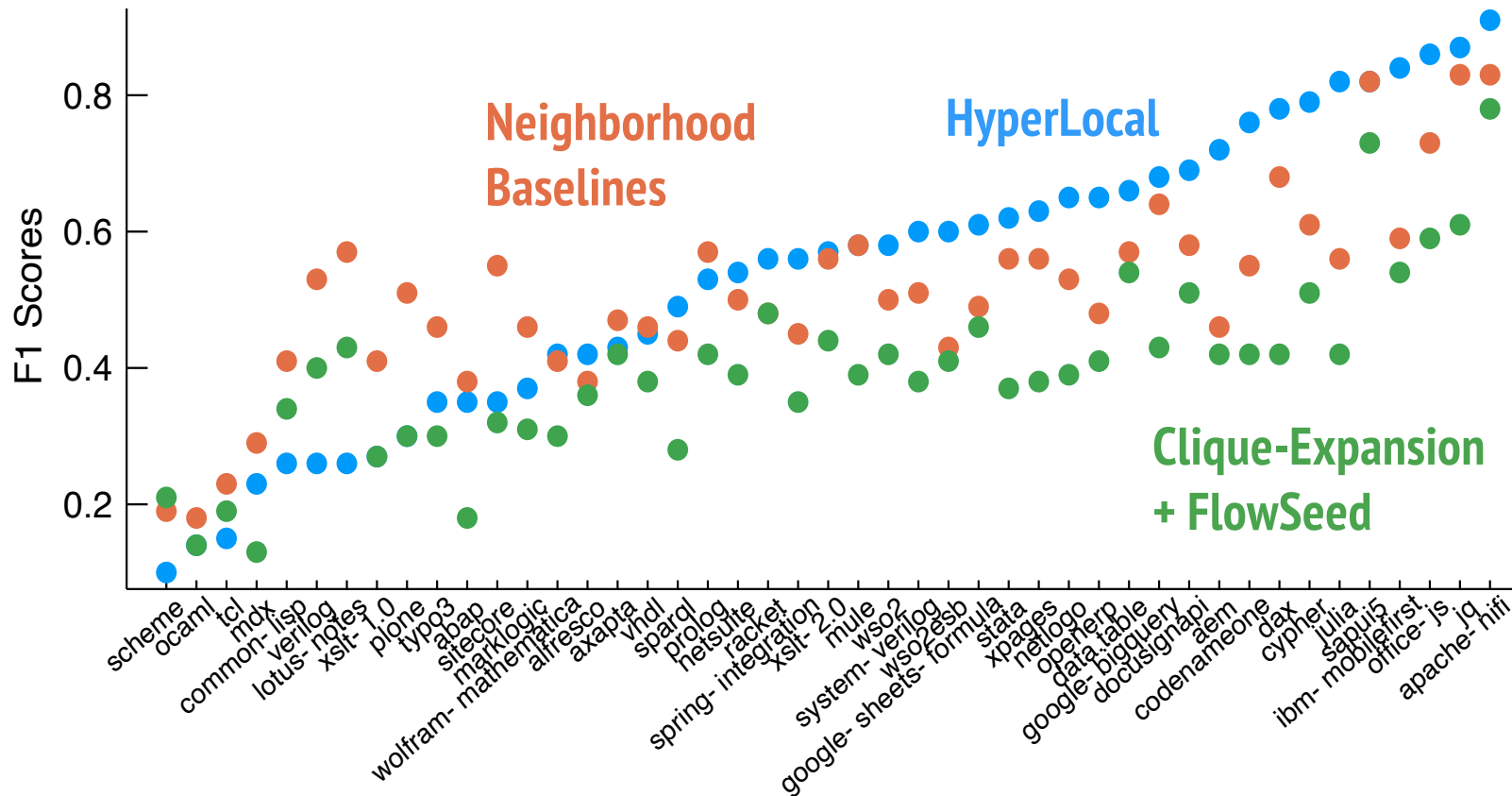
If a target set $T \subset V$ satisfies

$$\frac{\text{vol}(T \cap R)}{\text{vol}(T)} \geq \frac{\text{vol}(\bar{T} \cap R)}{\text{vol}(\bar{T})} + \beta$$

If T overlaps enough with seed set R ...

Then $\phi(S) \leq \frac{1}{\beta} \phi(T)$ where S is the set returned by our algorithm.

...then our output has normalized cut almost as good as T .



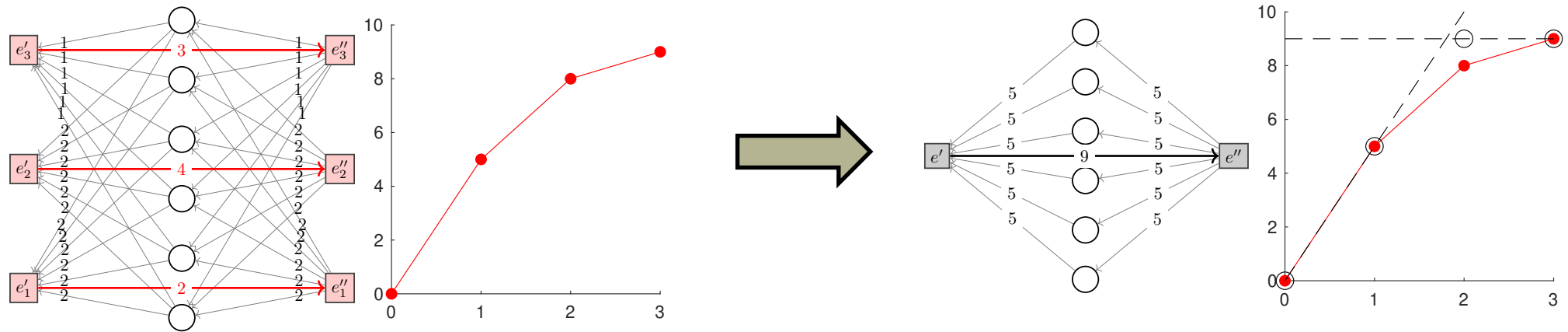
- 15M StackOverflow questions (nodes), answered by 1.1M users (hyperedges).
- mean hyperedge size 23.7, max hyperedge size ~ 60k.
- Tags provide ground truth cluster labels.
- Delta-linear splitting function $w_i = \min(i, 5000)$.

F1 recovery scores given a handful of nodes from the ground truth cluster T .

Cluster	$ T $	time (s)	HyperLocal	Baseline1	Baseline2
Amazon Fashion	31	3.5	0.83	0.77	0.6
All Beauty	85	30.8	0.69	0.60	0.28
Appliances	48	9.8	0.82	0.73	0.56
Gift Cards	148	6.5	0.86	0.75	0.71
Magazine Subscriptions	157	14.5	0.87	0.72	0.56
Luxury Beauty	1581	261	0.33	0.31	0.17
Software	802	341	0.74	0.52	0.24
Industrial & Scientific	5334	503	0.55	0.49	0.15
Prime Pantry	4970	406	0.96	0.73	0.36

- 2.3M Amazon products (nodes), reviewed by 4.3M users (hyperedges).
- mean hyperedge size > 17 , max hyperedge size $\sim 9.3k$.
- Product categories provide ground truth cluster labels.
- All-or-nothing penalty ($w_i = 1$).

Gadget reductions sometimes create dense graphs, which can make computations expensive.



Theorem [Veldt-Benson-Kleinberg 20c]. Any submodular C-B splitting function can be ε -approx with $\log r / \varepsilon$ splitting functions (instead of $r, r = \text{hyperedge size}$).

And one specific case...

- $r = 60k$ clique expansion only need $O(r / \sqrt{\varepsilon})$ instead of $O(r^2)$

We can now model and use hypergraph min s-t cuts.

1. A model for hypergraph cuts.

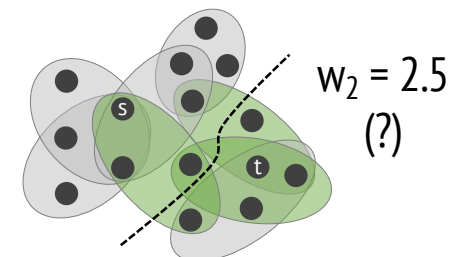
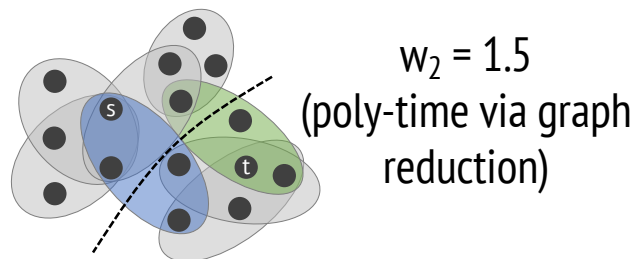
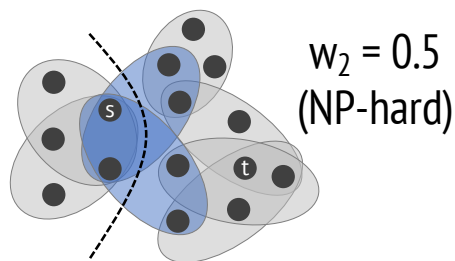
C-B splitting functions that depend on # of nodes on small side of the cut


2. Algorithm for min s-t cuts with submodular C-B splitting functions.

Graph-reducible if and only if C-B splitting function is submodular

3. Applications to local hypergraph clustering.

Strong locality lets us scale to large hypergraphs with large hyperedges



- Hypergraph Cuts with General Splitting Functions. Nate Veldt, Austin R. Benson, and Jon Kleinberg. arXiv:2001.02817, 2020.
- Localized Flow-Based Clustering in Hypergraphs. Nate Veldt, Austin R. Benson, and Jon Kleinberg. Proc. Of KDD, 2020.  github.com/nveldt/HypergraphFlowClustering
- Augmented Sparsifiers for Generalized Hypergraph Cuts. Nate Veldt, Austin R. Benson, and Jon Kleinberg. arXiv:2007.08075, 2020.



w/ M. Schaub, A. Jadbabaie,
G. Lippner, and P. Horn

Higher-order Network Data Analysis

1. Temporal evolution of higher-order interactions.
Simplicial Closure and Higher-order Link Prediction, PNAS 2018.
2. Clustering in large networks of higher-order interactions.
Minimizing Localized Ratio Cuts in Hypergraphs, KDD, 2020.
3. Diffusions over higher-order interactions in networks.
Random walks on simplicial complexes and the normalized Hodge 1-Laplacian, SIAM Review, 2020.

Background. Graph Laplacians, diffusions, and spectral graph theory underly many graph data methods.

D = diagonal degree matrix, A = adjacency matrix, $L = D - A$ is graph Laplacian..

Low-dimensional embeddings

[Belkin-Niyogi 02; Coifman-Lafon 06]

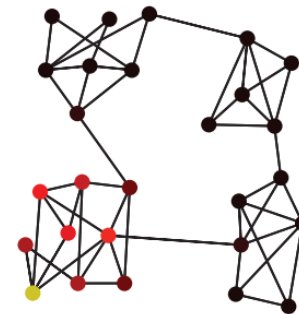


$$N \begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} \lambda_1 x_1 & \lambda_2 x_2 \end{bmatrix}$$

Norm. Lap. $N = D^{-1/2} L D^{-1/2}$.

Personalized PageRank

[Andersen-Chung-Lang 08; Gleich 15]



$$(\beta I + L D^{-1}) x = v$$

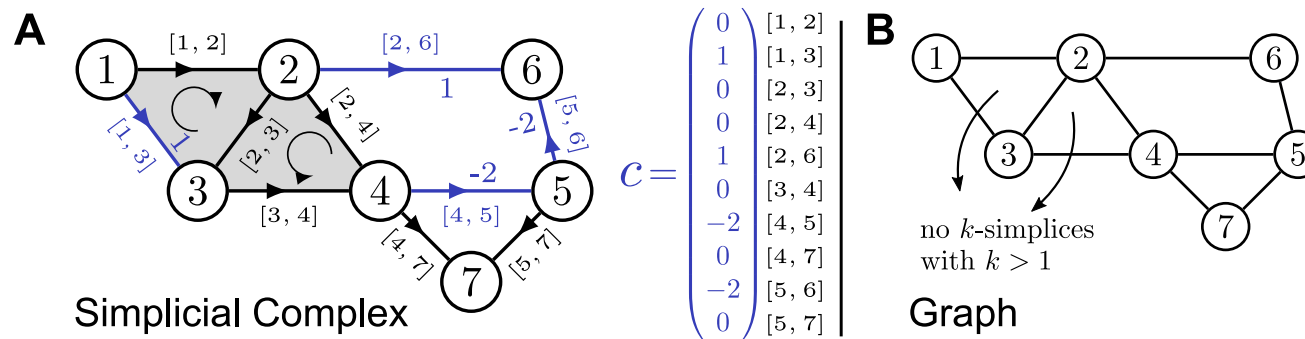
Random walk Lap. $L D^{-1}$.

What is a “higher-order” Laplacian?

Background. By interpreting our data as a simplicial complex, we can get higher-order Laplacians.

See *Hodge Laplacians on Graphs* by Lek-Heng Lim.

- (Abstract) simplicial complex X : if $A \in X$ and $B \subseteq A$, then $B \in X$.
- Graph $G = (V, E)$ as a simplicial complex: $X = V \cup E$.
- Can induce a complex from higher-order interactions.



$L_0 = 0^T 0 + B_1 B_1^T = D - A$ graph Laplacian operates on nodes

$L_1 = B_1^T B_1 + B_2 B_2^T$ Hodge Laplacian operates on *oriented* edges

B_1 maps edges to nodes, B_2 maps triangles to edges.

We spent a lot of time getting the normalization and connections to random walks right.

THEOREM 3.4 (stochastic lifting of the normalized Hodge 1-Laplacian). *The matrix $-\mathcal{L}_1/2$ has a stochastic lifting, i.e., there exists a column stochastic matrix $\hat{\mathbf{P}}$ such that $-\frac{1}{2}\mathcal{L}_1\mathbf{V}^\top = \mathbf{V}^\top\hat{\mathbf{P}}$. Specifically, $\hat{\mathbf{P}} := \frac{1}{2}\mathbf{P}_{\text{lower}} + \frac{1}{2}\mathbf{P}_{\text{upper}}$, where $\mathbf{P}_{\text{lower}}$ is the transition matrix of a random walk determined by the lower-adjacent connections and $\mathbf{P}_{\text{upper}}$ is the transition matrix of a random walk determined by the upper-adjacent connections. The transition matrix $\mathbf{P}_{\text{lower}}$ is defined by a “forward walk” and a “backward walk” component moving in the orientation of the edges or against it, respectively:*

$$(3.9) \quad \mathbf{P}_{\text{lower}} := \frac{1}{2} (\mathbf{P}_{\text{lower,forward}} + \mathbf{P}_{\text{lower,backward}}),$$

$$(3.10) \quad \mathbf{P}_{\text{lower,forward}} = \mathbf{M}_f \text{diag}(\mathbf{M}_f \mathbf{1})^{-1},$$

$$(3.11) \quad \mathbf{P}_{\text{lower,backward}} = \mathbf{M}_b \text{diag}(\mathbf{M}_b \mathbf{1})^{-1},$$

where $\mathbf{M}_f = \hat{\mathbf{D}}_2(\hat{\mathbf{B}}_1^-)^\top \hat{\mathbf{B}}_1^+$ and $\mathbf{M}_b = \hat{\mathbf{D}}_2(\hat{\mathbf{B}}_1^+)^\top \hat{\mathbf{B}}_1^-$ are (weighted) lower-adjacency matrices corresponding to forward and backward walks along the edges (see Lemma 3.2) and $\hat{\mathbf{D}}_2 = \text{diag}(\mathbf{D}_2, \mathbf{D}_2)$. The transition matrix $\mathbf{P}_{\text{upper}}$ describes a random walk along upper-adjacent faces as follows:

$$(3.12) \quad \mathbf{P}_{\text{upper}} = \hat{\mathbf{A}}_u \hat{\mathbf{D}}_4^{-1} + \frac{1}{2} \begin{pmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{pmatrix} \hat{\mathbf{D}}_5,$$

where $\hat{\mathbf{A}}_u = \hat{\mathbf{B}}_2^+(\hat{\mathbf{B}}_2^-)^\top + \hat{\mathbf{B}}_2^-(\hat{\mathbf{B}}_2^+)^\top$ is the matrix of upper-adjacent connections as defined in Lemma 3.2 and $\hat{\mathbf{D}}_4$ is a diagonal matrix:

$$(3.13) \quad (\hat{\mathbf{D}}_4)_{[i,j],[i,j]} = \begin{cases} 1 & \text{if } \deg([i,j]) = 0, \\ 3 \cdot \deg([i,j]) & \text{otherwise.} \end{cases}$$

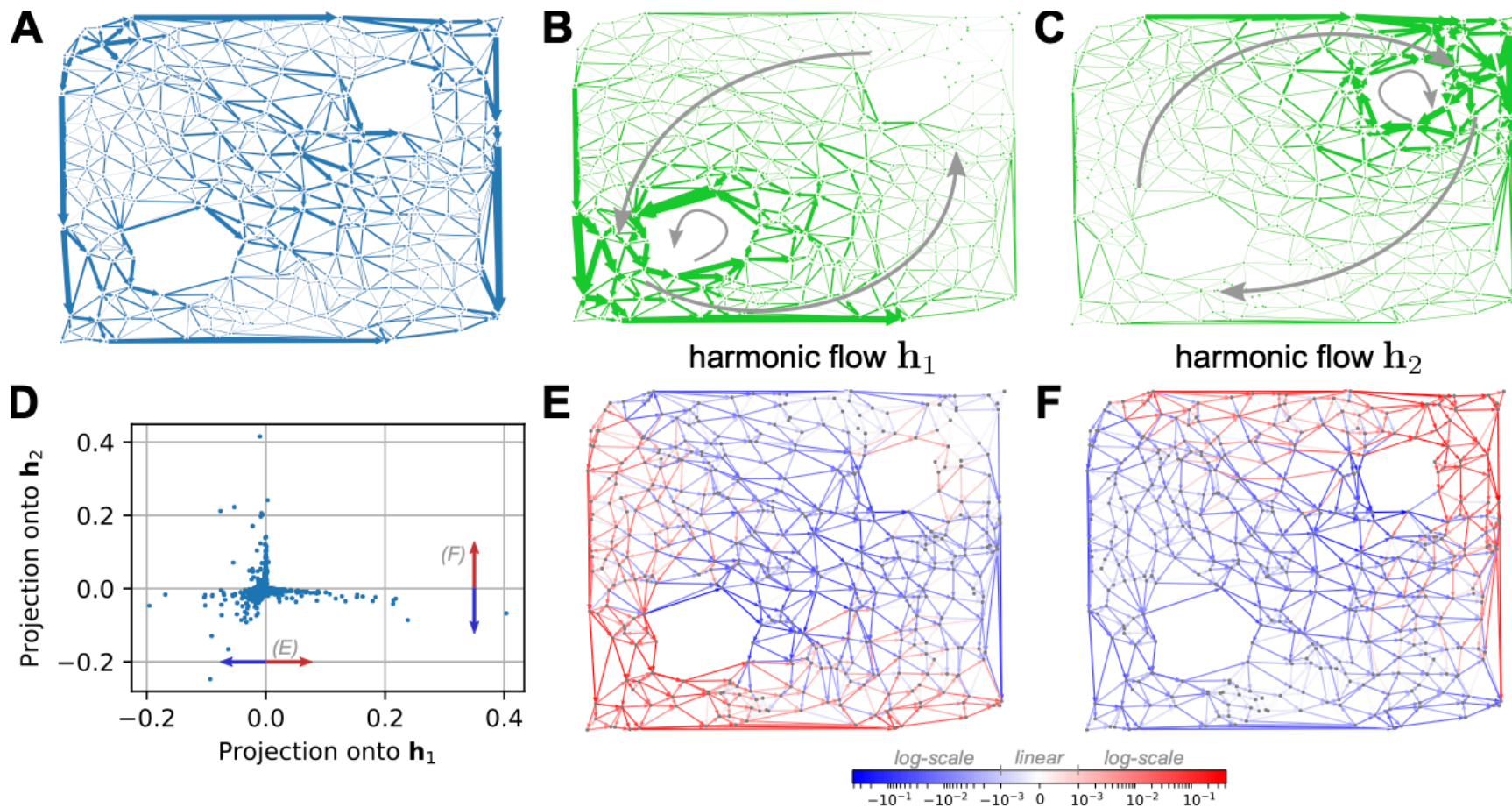
Here $\hat{\mathbf{D}}_5$ is the diagonal matrix selecting all edges with no upper-adjacent faces:

$$(3.14) \quad (\hat{\mathbf{D}}_5)_{[i,j],[i,j]} = \begin{cases} 1 & \text{if } \deg([i,j]) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Random Walks on Simplicial Complexes and the normalized Hodge 1-Laplacian.
Michael T. Schaub, Austin R. Benson, Paul Horn, Gabor Lippner, and Ali Jadbabaie.
SIAM Review, 2020.

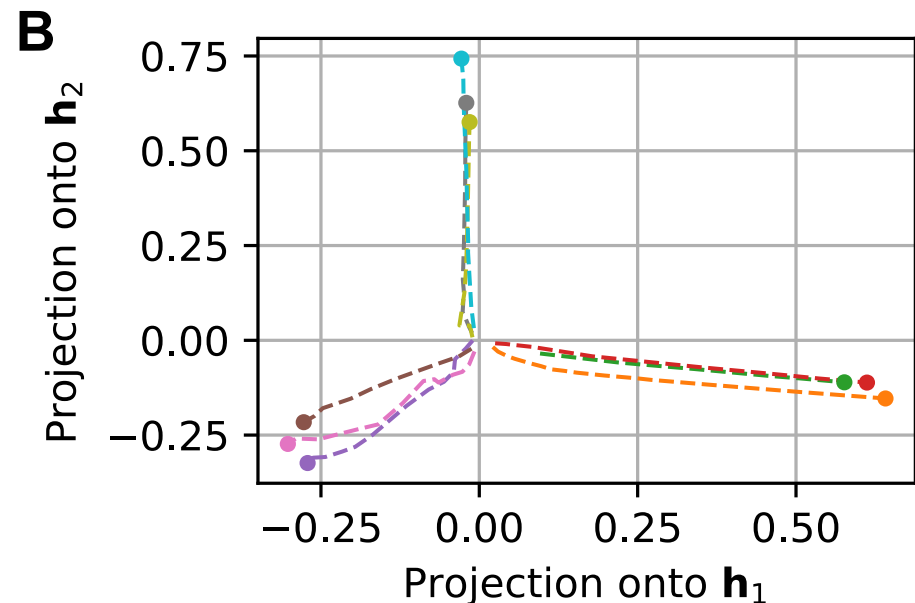
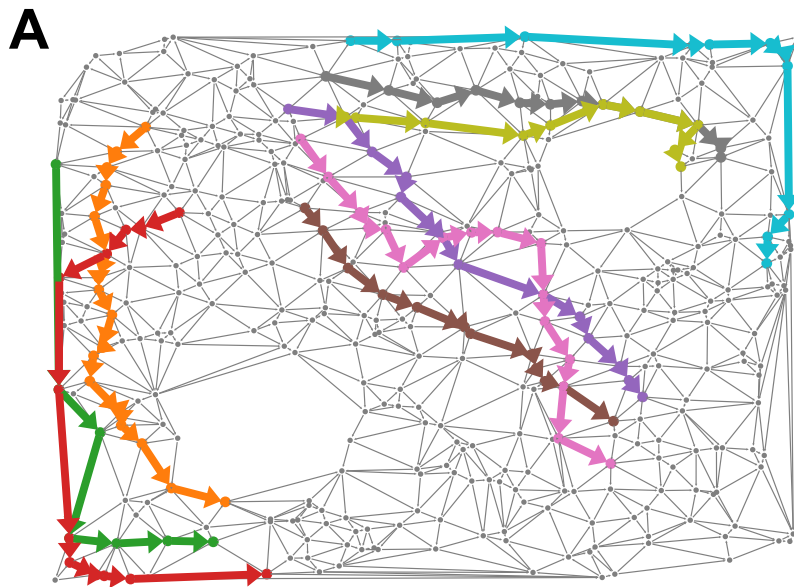
Flow embeddings are the higher-order analog of diffusion maps.

$N_1 \begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} \lambda_1 x_1 & \lambda_2 x_2 \end{bmatrix}$ for normalized Hodge 1-Laplacian N_1 .



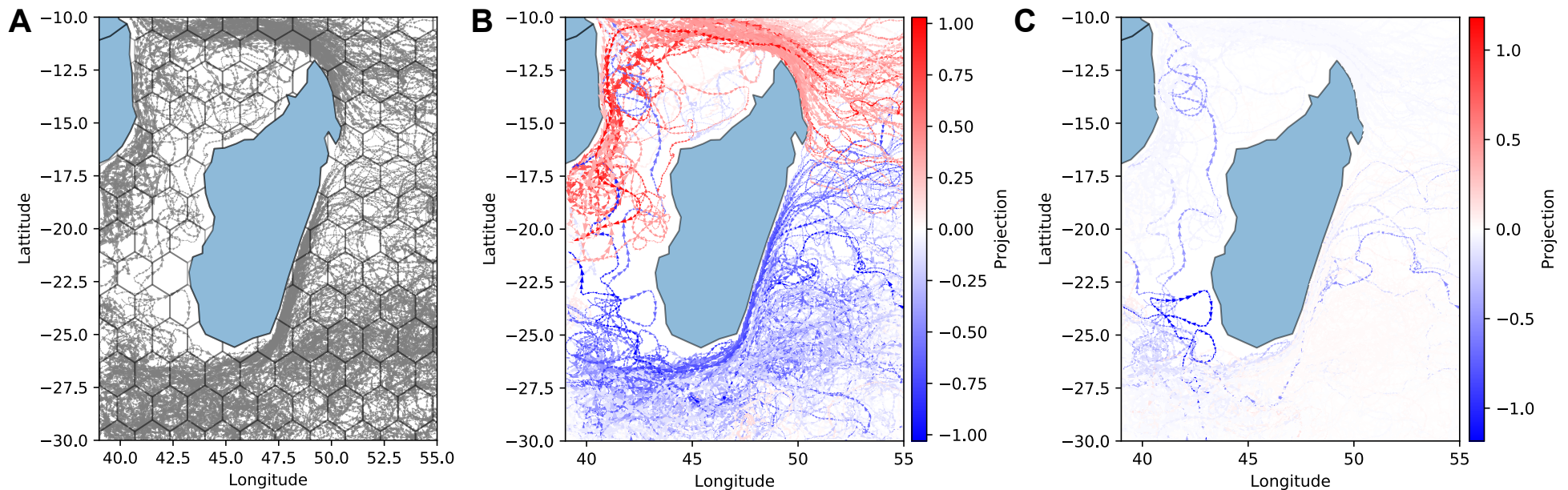
Flow embeddings are the higher-order analog of diffusion maps.

$N_1 \begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} \lambda_1 x_1 & \lambda_2 x_2 \end{bmatrix}$ for normalized Hodge 1-Laplacian N_1 .

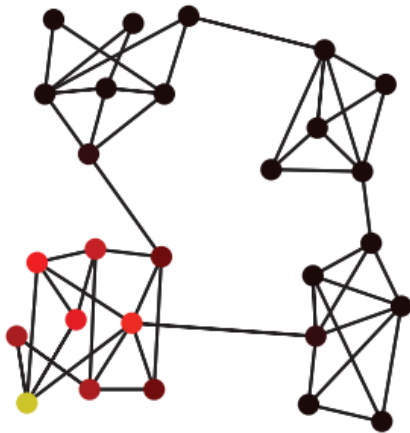


Flow embeddings are the higher-order analog of diffusion maps.

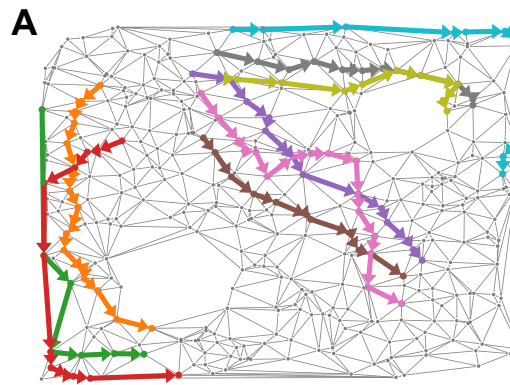
$N_1 \begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} \lambda_1 x_1 & \lambda_2 x_2 \end{bmatrix}$ for normalized Hodge 1-Laplacian N_1 .



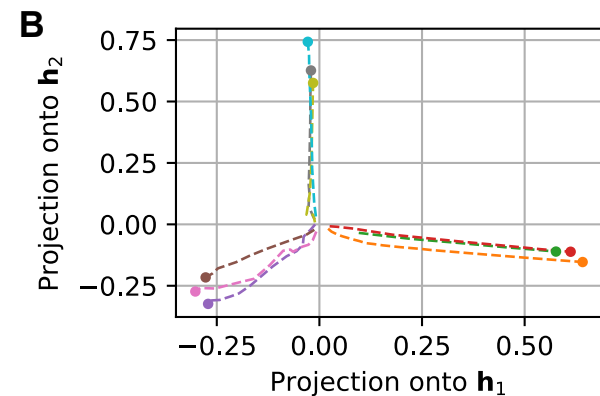
The holes correspond to the idea of homology in algebraic topology.



First eigenvectors of the **graph Laplacian** capture (near) connected components, or zeroth-order homology.

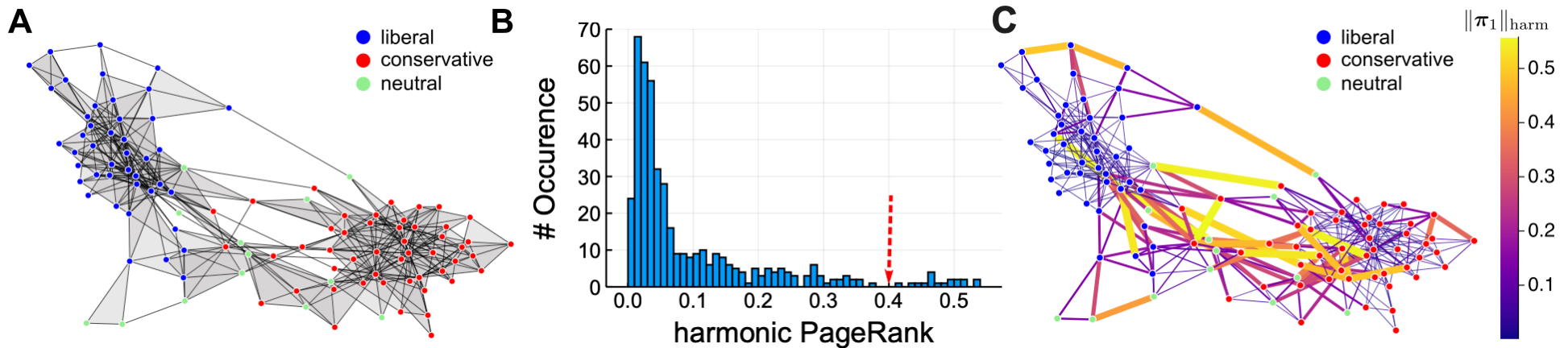


First eigenvectors of the **Hodge Laplacian** capture (near) topological holes, or first-order homology



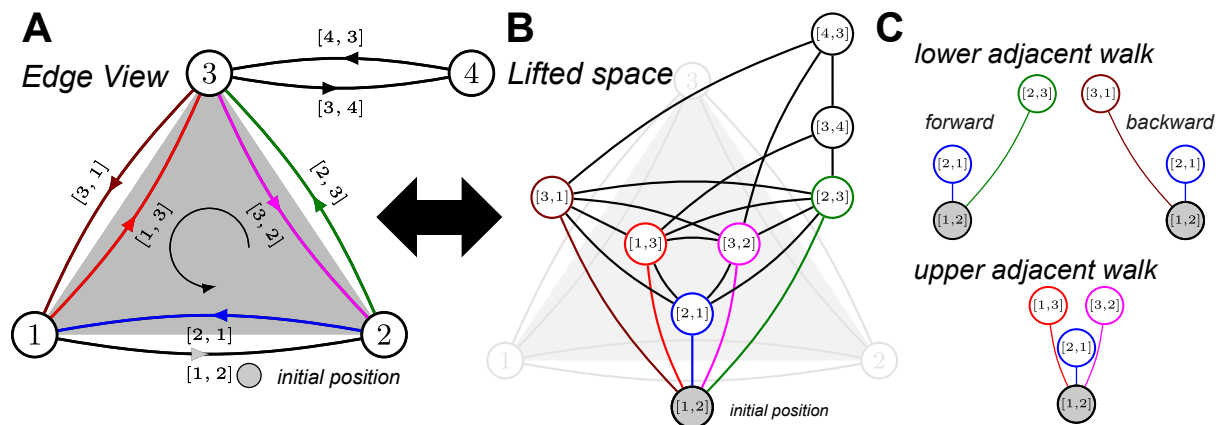
We also have simplicial Personalized PageRank.


$(\beta I + \mathcal{L}_1)x = v$ for (asymmetric) normalized Hodge 1-Laplacian \mathcal{L}_1 .



Abstract simplicial complexes are another way to model and analyze higher-order network data.

1. Algebraic topology provides the computational framework.
2. The hard part is getting a normalization scheme that connects the Hodge Laplacian to diffusions and “respects the topology.”
3. We can apply these ideas to graph algorithms based on random walks.



- Random Walks on Simplicial Complexes and the normalized Hodge 1-Laplacian. Michael T. Schaub, Austin R. Benson, Paul Horn, Gabor Lippner, and Ali Jadbabaie. SIAM Review, 2020.
- Graph-based Semi-Supervised & Active Learning for Edge Flows. Junteng Jia, Michael T. Schaub, Santiago Segarra, and Austin R. Benson. Proc. of KDD, 2019.  github.com/000Justin000/ssl_edge

Computational frameworks for higher-order data analysis.

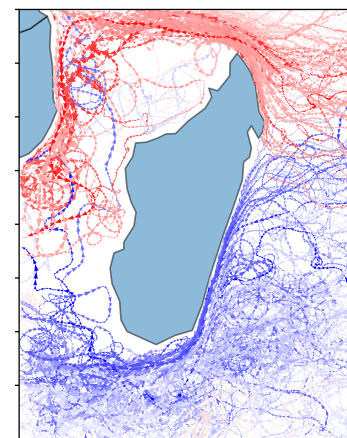
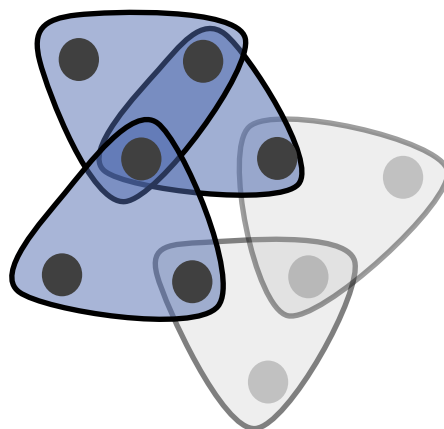
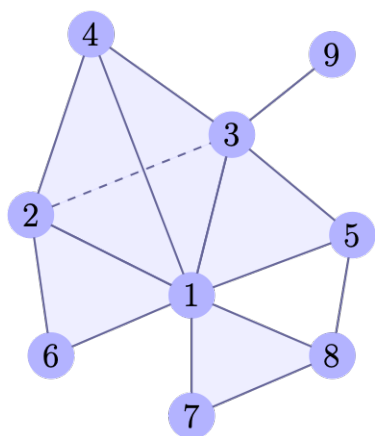
THANKS! Austin R. Benson

Slides. bit.ly/arb-TAMU-20

<http://cs.cornell.edu/~arb>

 @austinbenson

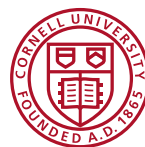
 arb@cs.cornell.edu



Lots of data available at <https://www.cs.cornell.edu/~arb/data/>



CHASE 



Cornell University

Supported by ARO MURI, ARO Award W911NF19-1-0057, NSF Award DMS-1830274, and JP Morgan Chase & Co.