Higher-order network science (with hypergraph cuts)

Austin Benson · Cornell University

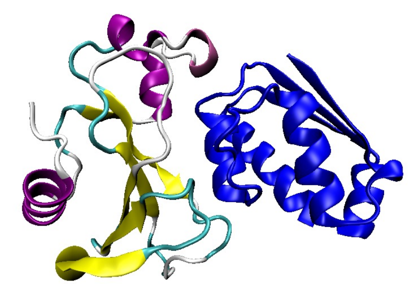
WPI CS Colloquium · May 7, 2021

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



Cell biology
nodes are proteins
edge between two proteins that interact

Network science 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)

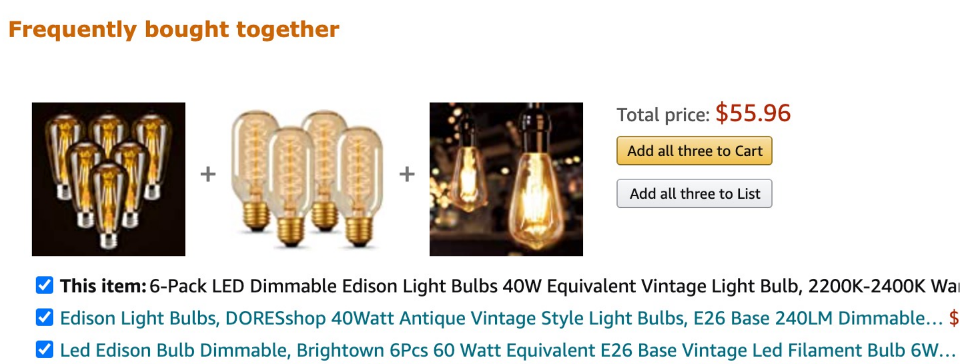
4. **Ranking**
   Which things are important? (PageRank and its variants)

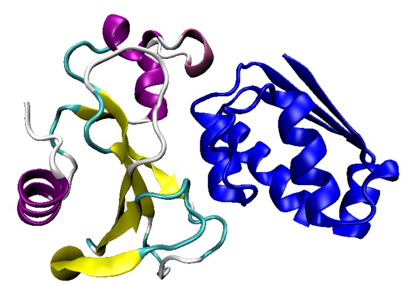
Real-world systems are composed of "higher-order" interactions that we often reduce to pairwise ones.

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

Physical proximity
nodes are people
people gather in groups

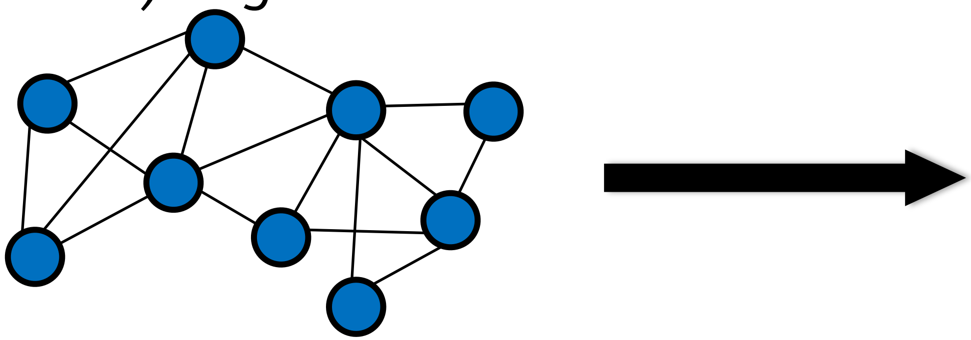
Commerce
nodes are products
several products can be purchased at once

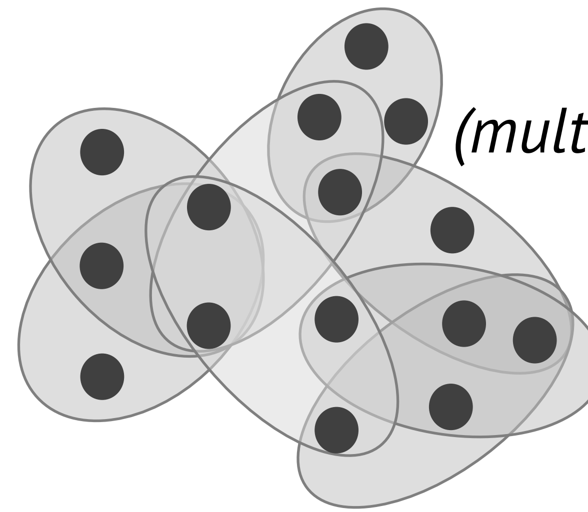




Cell biology
nodes are proteins
protein complexes may involve several proteins

Graph with (pairwise) edges





Hypergraph with (multiway) hyperedges

What new insights does this give us?

We can ask the same network science questions while accounting for 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)

Graphs are a great mod… …nships.
But hypergr… …der" structure.

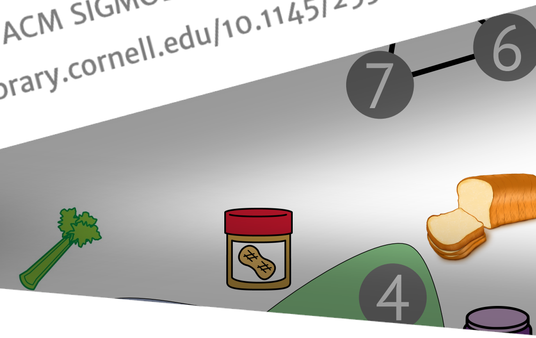
ARTICLE

Dynamic itemset counting and implication rules for market basket data

Authors: Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, Shalom Tsur Authors Info & Affiliations

Publication: SIGMOD '97: Proceedings of the 1997 ACM SIGMOD international conference on Management of data • June 1997 • Pages 255–264 • https://doi-org.proxy.library.cornell.edu/10.1145/253260.253325

…is information, how should a grocery store separate products into different departments or aisles? Or bundle their products?



Having access to "higher-order" data …affect your decision!

Retail pr…

Hyperedge…

items t…

p…

relationship among a set of objects (n…)

…{1,2,3} …?

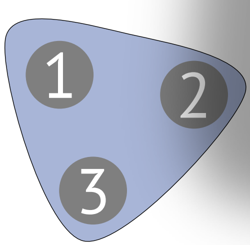
March 2020

SHOPPER: A probabilistic model of consumer choice with substitutes and complements

Francisco J. R. Ruiz, Susan Athey, David M. Blei

Translator D…

Ann. Appl. Stat. 14(1): 1-27 (March 2020). DOI: 10.1214/19-AOAS1265



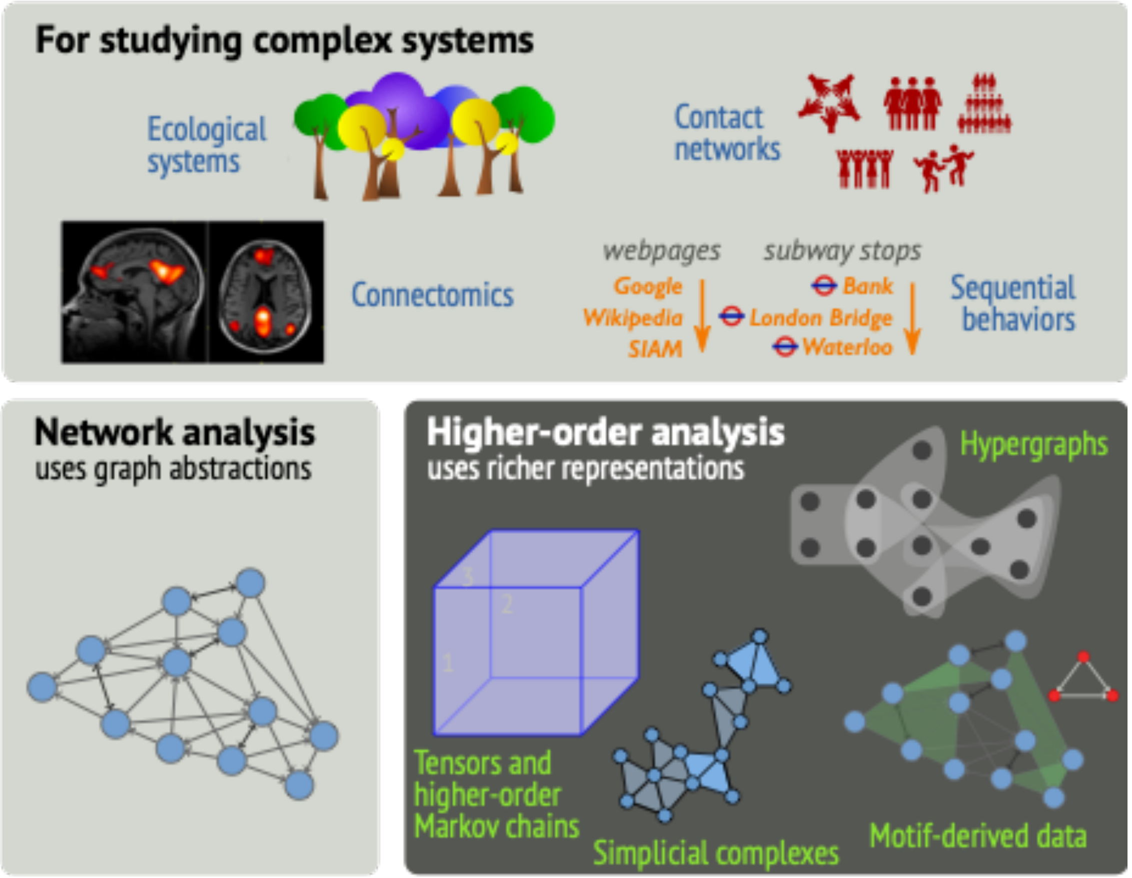
Research | January 21, 2021

Print

Higher-order Network Analysis Takes Off, Fueled by Old Ideas and New Data

By Austin R. Benson, David F. Gleich, and Desmond J. Higham

arXiv:2103.05031



Higher-order network science

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

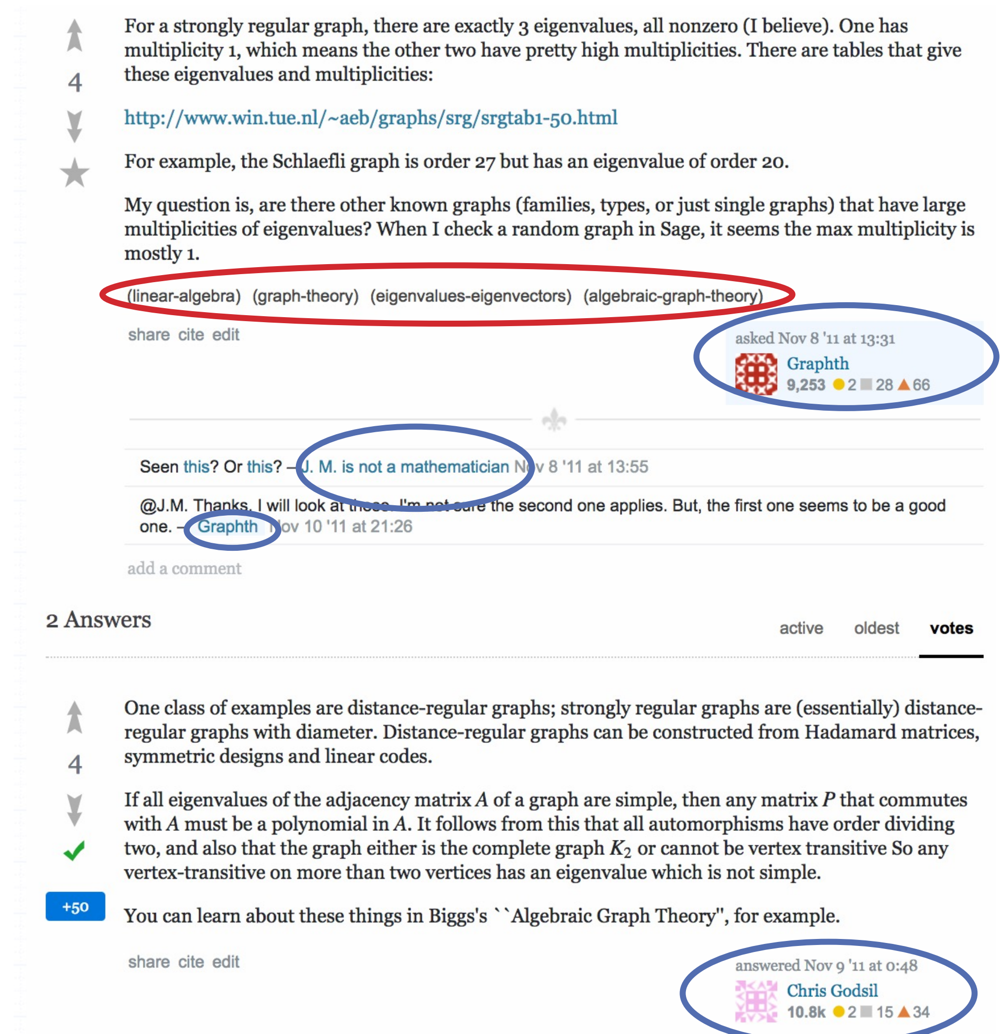
1. Temporal evolution of higher-order interactions.
   Simplicial Closure and Higher-order Link Prediction, PNAS 2018.

2. Hypergraph cuts for local and global clustering.
   Hypergraph Cuts with General Splitting Functions, arXiv, 2020.
   Minimizing Localized Ratio Cuts in Hypergraphs, KDD, 2020.
   Hypergraph clustering: from blockmodels to modularity, arXiv, 2021.

We collected many datasets of timestamped hyperedges

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.



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.

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

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

$t_3 : \{1, 6\}$

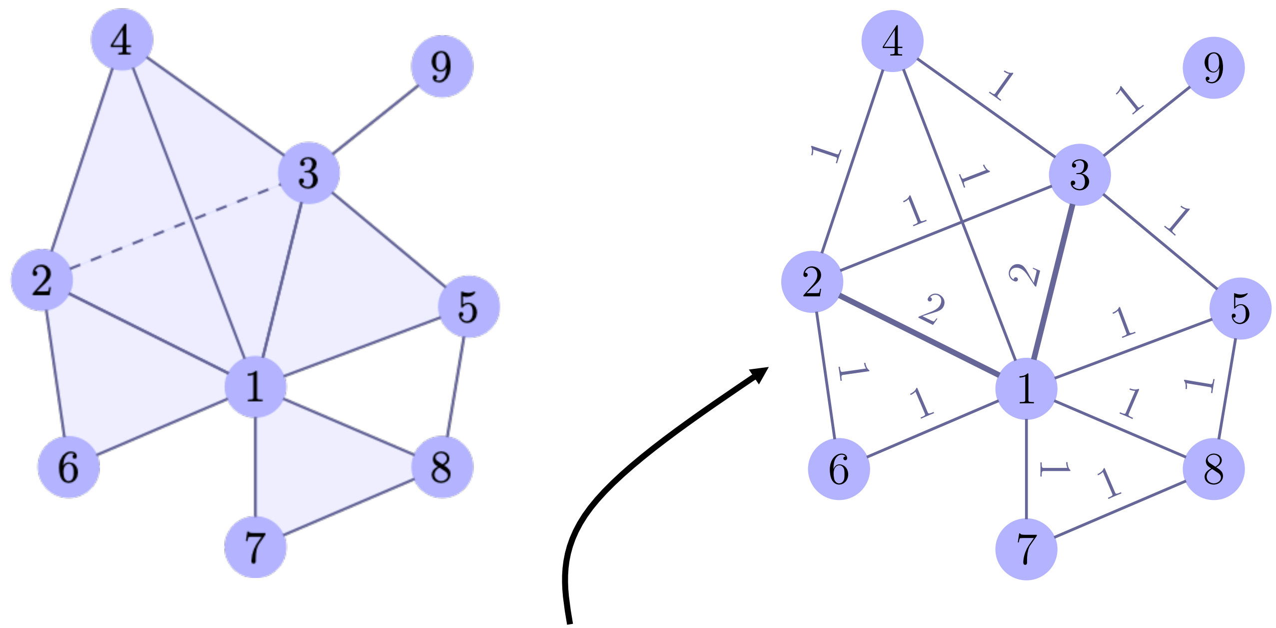
$t_4 : \{2, 6\}$

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

$t_6 : \{3, 9\}$

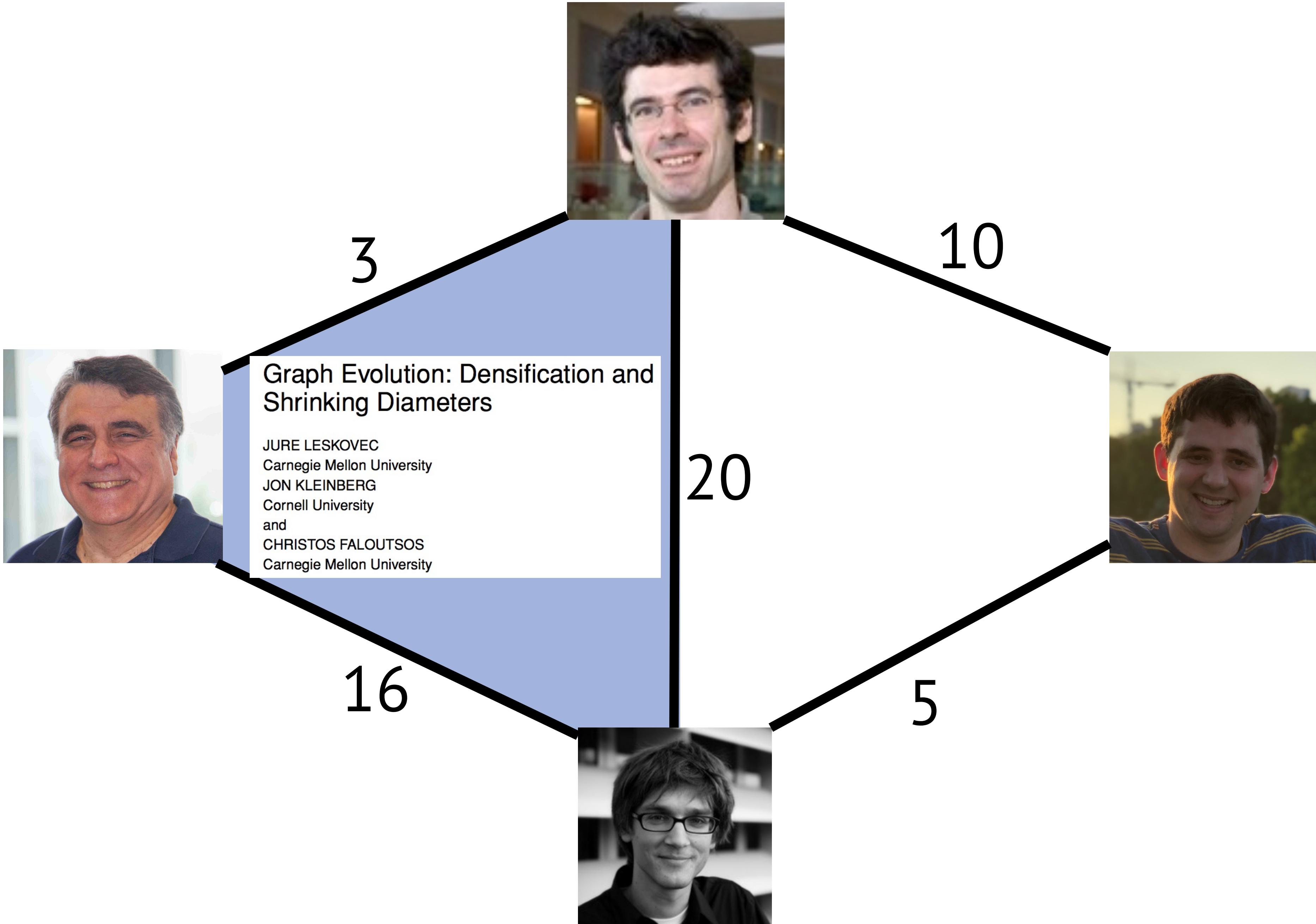
$t_7 : \{5, 8\}$

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

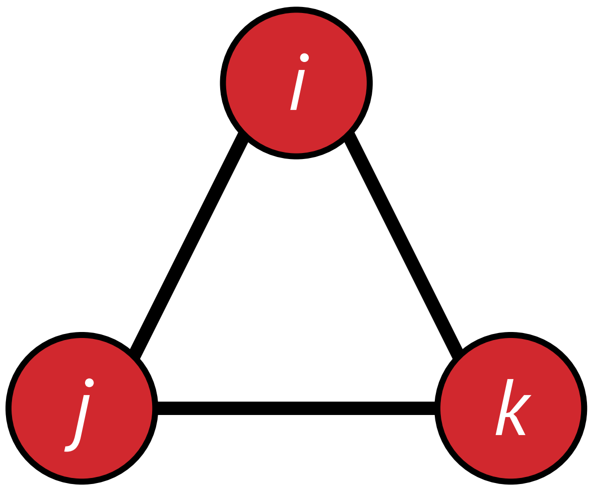


Projected graph ***W***.

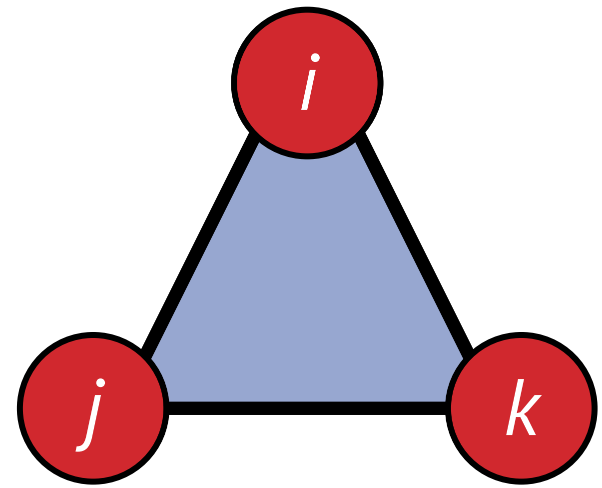
W_{ij} = # of hyperedges containing nodes i and j.


3
10
20
16
5
Graph Evolution: Densification and Shrinking Diameters
JURE LESKOVEC
Carnegie Mellon University
JON KLEINBERG
Cornell University
and
CHRISTOS FALOUTSOS
Carnegie Mellon University

What's more common in empirical data?



or



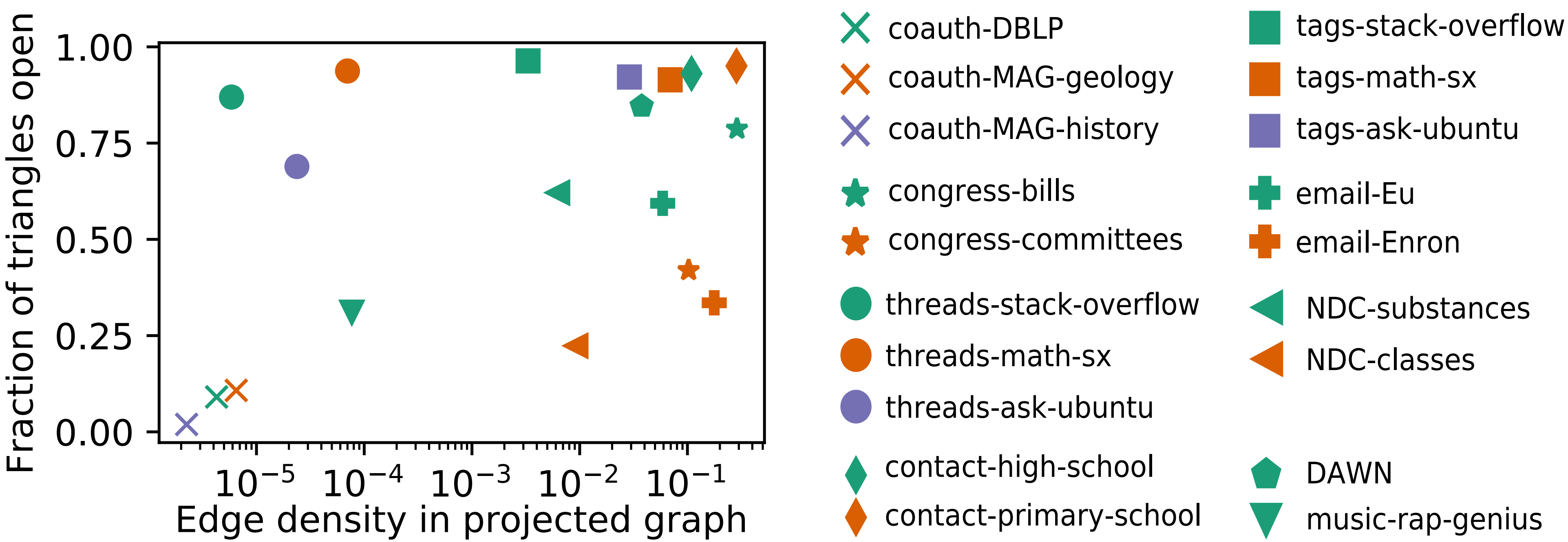
Open triangle

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

Closed triangle

there is some hyperedge 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.

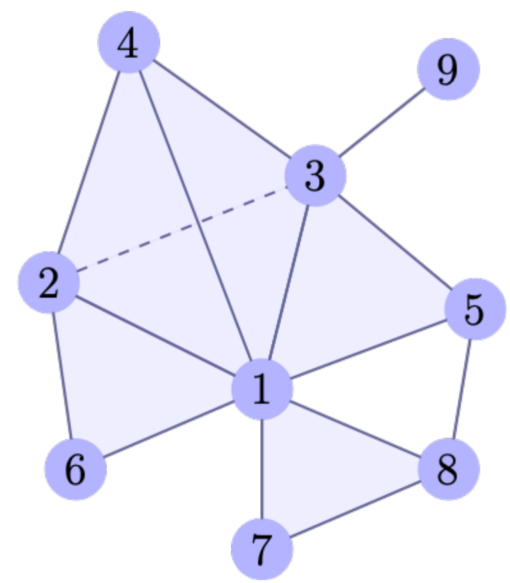


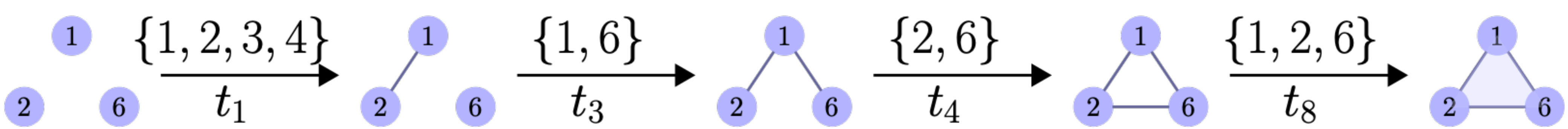
See also Patania-Petri-Vaccarino (2017) for similar ideas in collaboration networks.

How do new closed triangles appear?
Can we predict which new higher-order interactions will occur?

Triangles "fill in" or "close" over time.

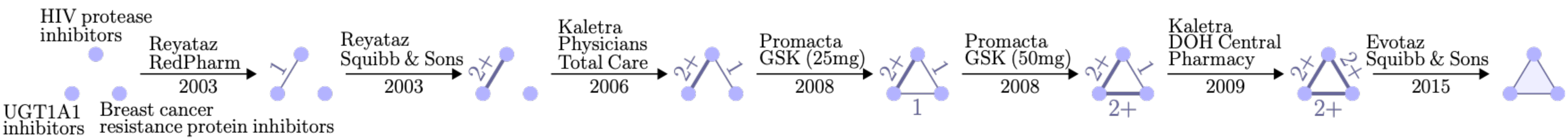
$t_1 : \{1, 2, 3, 4\}$
$t_2 : \{1, 3, 5\}$
$t_3 : \{1, 6\}$
$t_4 : \{2, 6\}$
$t_5 : \{1, 7, 8\}$
$t_6 : \{3, 9\}$
$t_7 : \{5, 8\}$
$t_8 : \{1, 2, 6\}$





Weak and strong ties are useful characterizations.

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

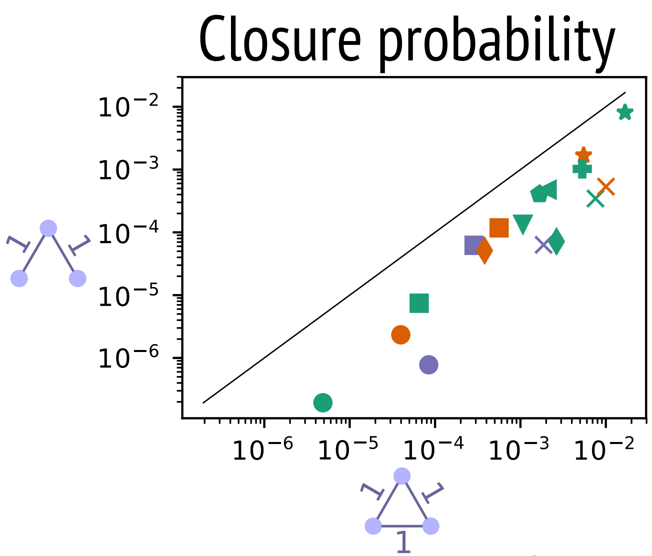


Bin weighted edges into “weak” and “strong ties” in the projected graph ***W***.
W_{ij} = # of simplices containing nodes *i* and *j*.

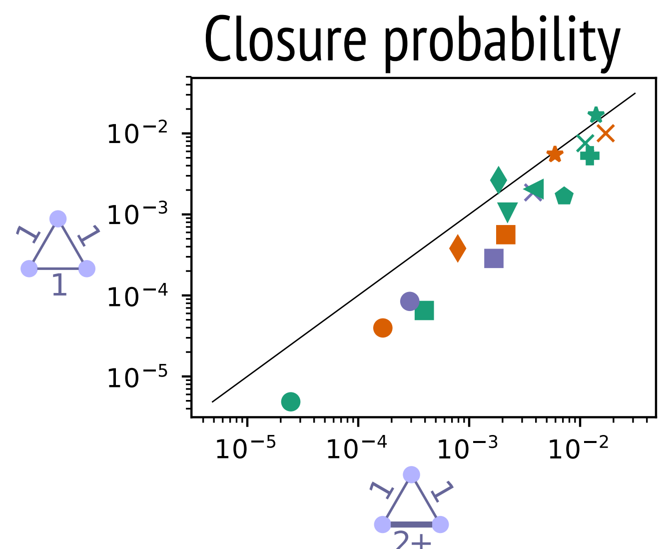
- **Weak ties.** $W_{ij} = 1$ (one hyperedge contains *i* and *j*)
- **Strong ties.** $W_{ij} \geq 2$ (at least hyperedges contain *i* and *j*)

Closure depends on structure in projected graph.

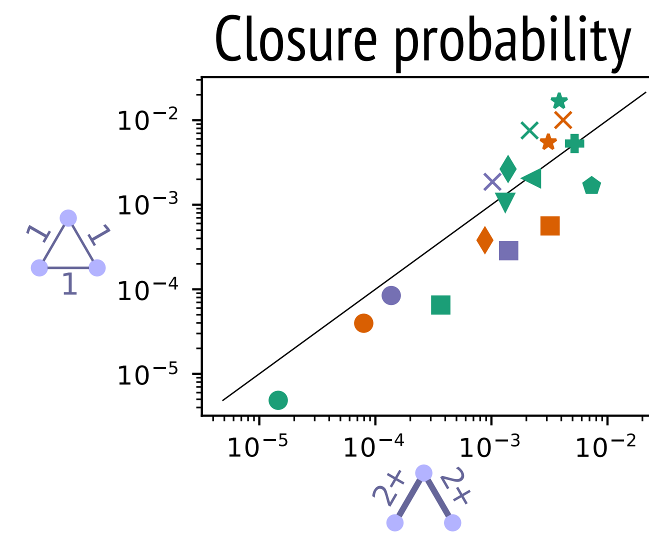
- First 80% of the data (in time) ⟶ record configurations of triplets not in closed triangle.
- Remainder of data ⟶ find fraction that are now closed triangles.



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]

We used this for a new higher-order link prediction task.

Data.

time

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

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

$t_3 : \{1, 6\}$

$t_4 : \{2, 6\}$

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

$t_6 : \{3, 9\}$

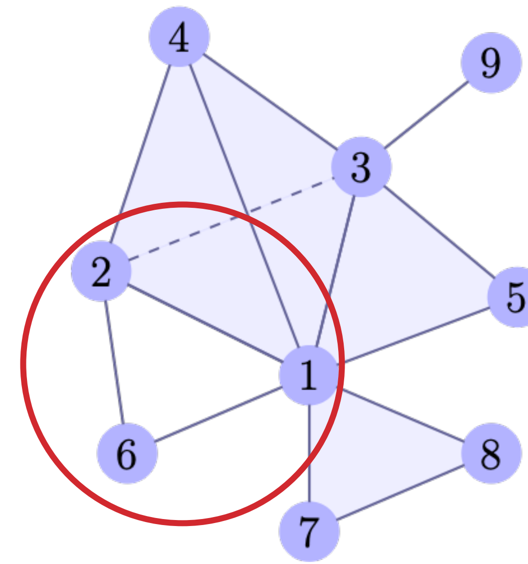
t

$t_7 : \{5, 8\}$

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

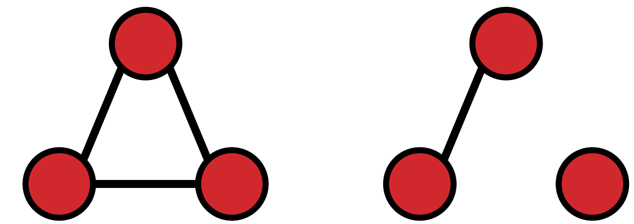
- Observe simplices up to time t.
- Predict which groups of > 2 nodes will appear after time t.

We predict structure that graph models would not even consider!



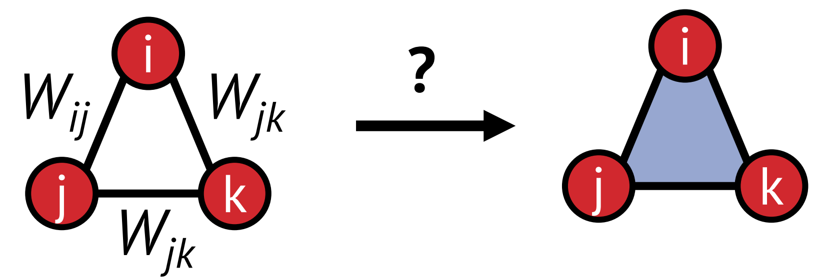
Our structural analysis helps us with prediction.

1. **Edge density matters.**
   Predicting open triangles first is the easiest way to get a performance boost.



2. **Tie strength matters a lot!**
   We tried lots of stuff, but simple averaging works extremely well.

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



In contrast…
Long paths help with classical link prediction methods [Liben-Nowell & Kleinberg 07]
Complex k-hop neighborhood computations work well for modern graph neural networks

Higher-order network science

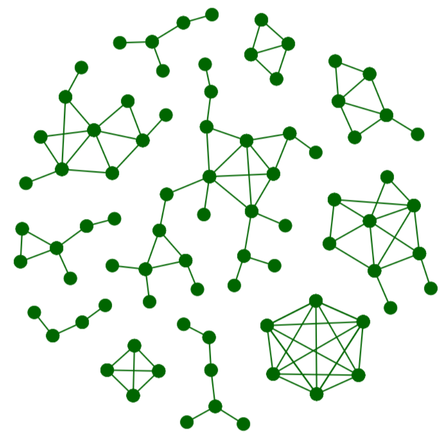
w/ N. Veldt, J. Kleinberg, P. Chodrow

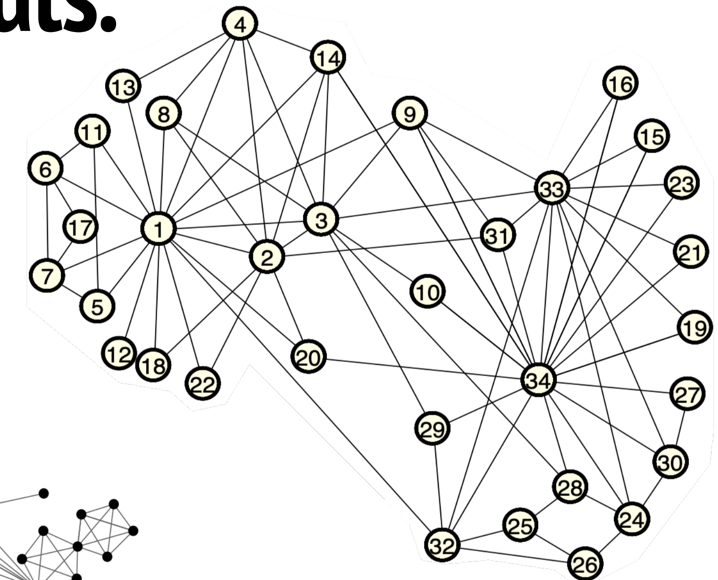
1. Temporal evolution of higher-order interactions.
   Simplicial Closure and Higher-order Link Prediction, PNAS 2018.

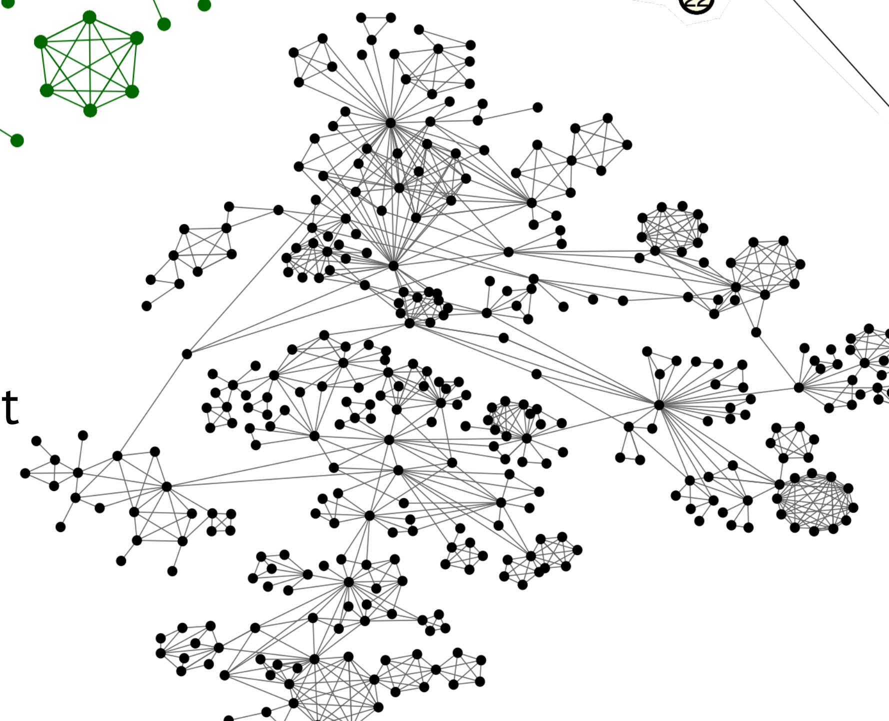
2. Hypergraph cuts for local and global clustering.
   Hypergraph Cuts with General Splitting Functions, arXiv, 2020.
   Minimizing Localized Ratio Cuts in Hypergraphs, KDD, 2020.
   Hypergraph clustering: from blockmodels to modularity, arXiv, 2021.

In network science, a wide array of applications rely on finding graph clusters and small graph cuts.

Applications
community detection
graph partitioning
semi-supervised learning
routing/flow problems
dense subgraph detection
localized clustering
⋮



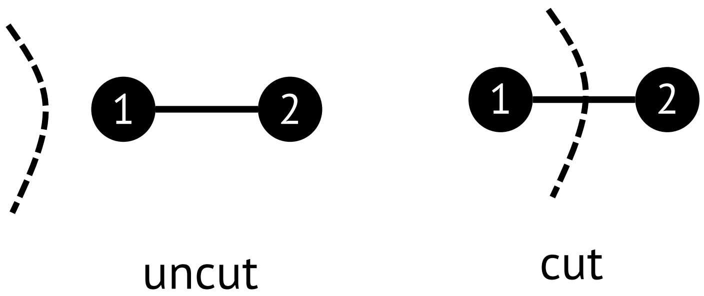




Cluster = densely connected node set that is sparsely connected to rest of graph

Cut = number of edges crossing a cluster boundary

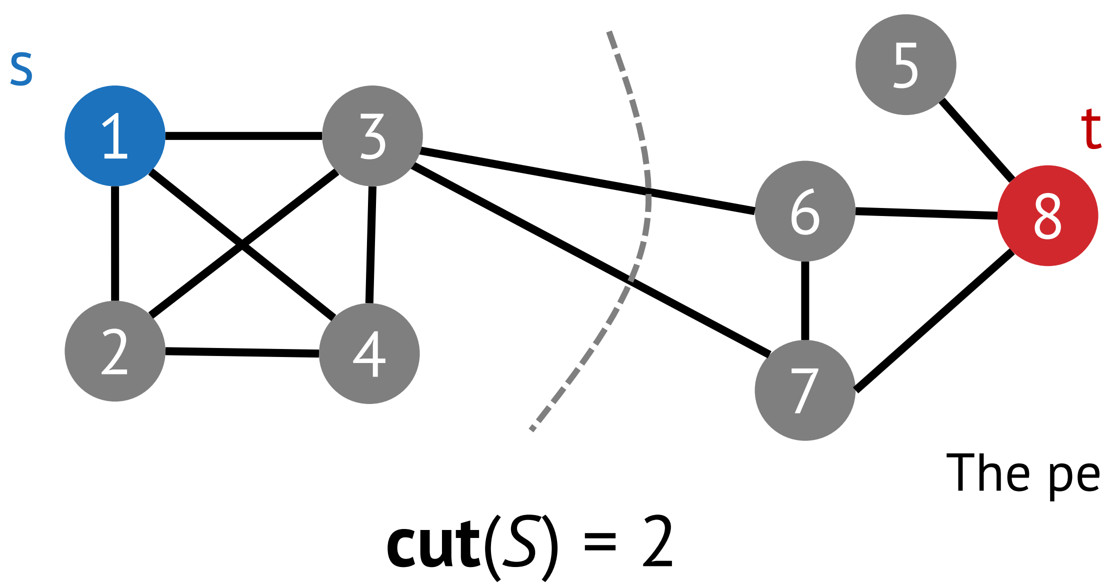
Cut and clustering problems are well understood and widely applied in graph analysis.



An edge is cut if its nodes are separated.

Types of cut problems
minimum s-t cut
multiway cut
min conductance cut
sparsest cut
⋮

A classical example of a graph cut problem is the minimum s-t cut



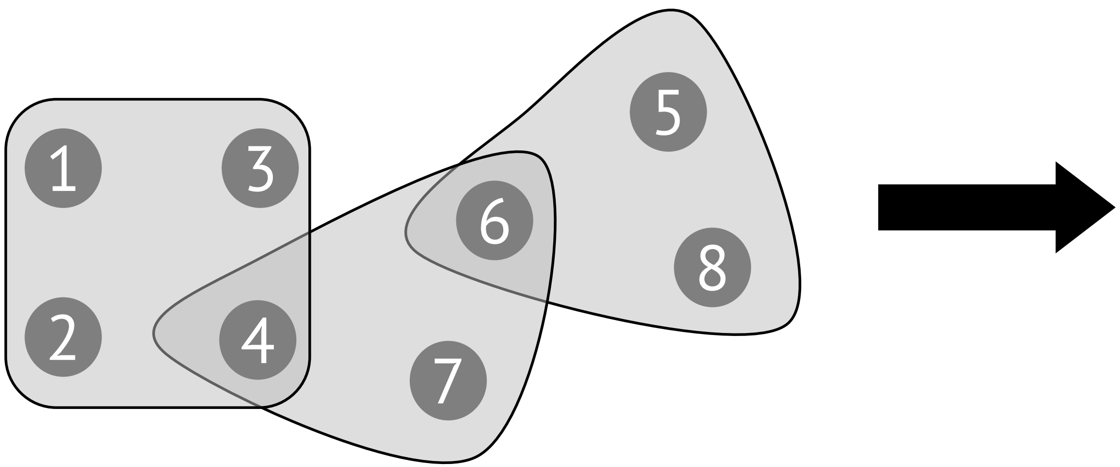
$$\text{minimize}_{S \subset V} \quad \text{cut}(S)$$
$$\text{subject to} \quad s \in S,\ t \notin S.$$

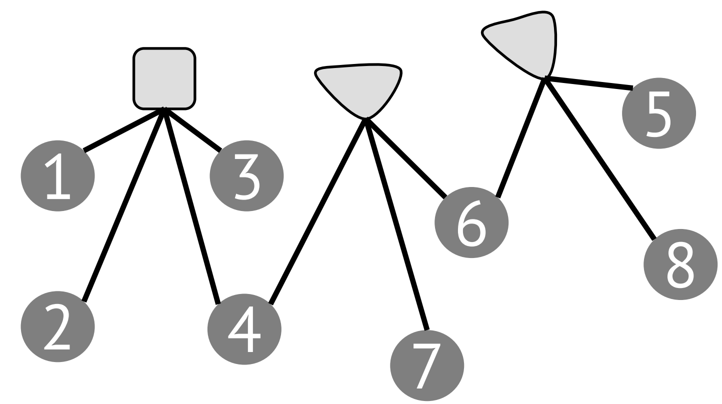
The penalty for cutting an edge is its weight.

How do we define hypergraph cut problems?
Once defined, how do we solve them?

A first thought.

Apply a bipartite graph expansion and solve the cut problem on the graph.





What exactly is this cut measuring? Is it right for applications? Are there alternatives?

There are two major challenges.

We'll address these with

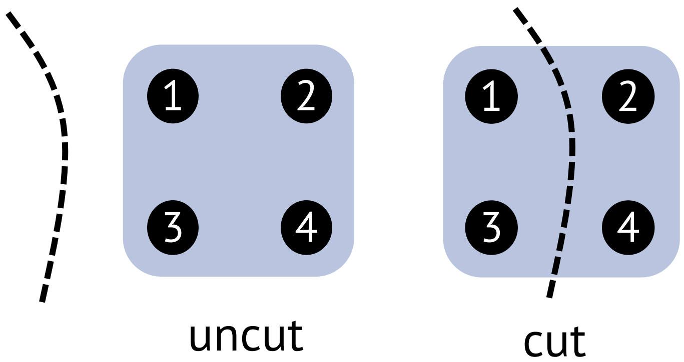
1. **Modeling Questions.** There are many ways to generalize graph cut and clustering problems to the hypergraph setting.

   Generalized and unifying models for hypergraph cut functions.

2. **Scalability Issues.** Hypergraphs can grow not only in terms of nodes and hyperedges, but also hyperedge size.

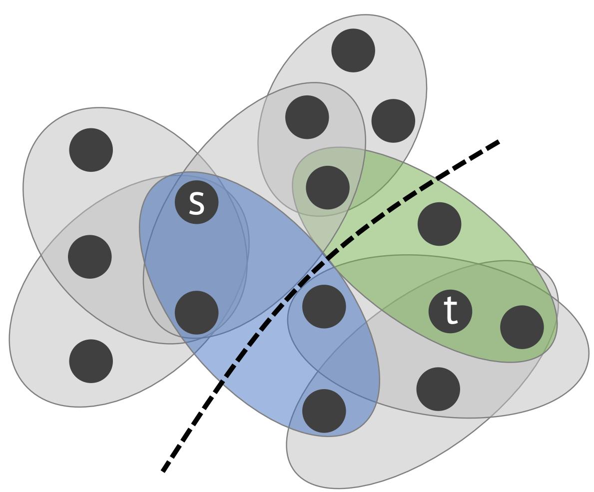
   Methods that scale up to hypergraphs with millions of nodes, and millions of ***very large*** *hyperedges.*

~~The~~ *A* hypergraph cut function has existed for decades.



A hyperedge is cut if its nodes are separated.

The hypergraph cut is the number of cut hyperedges.

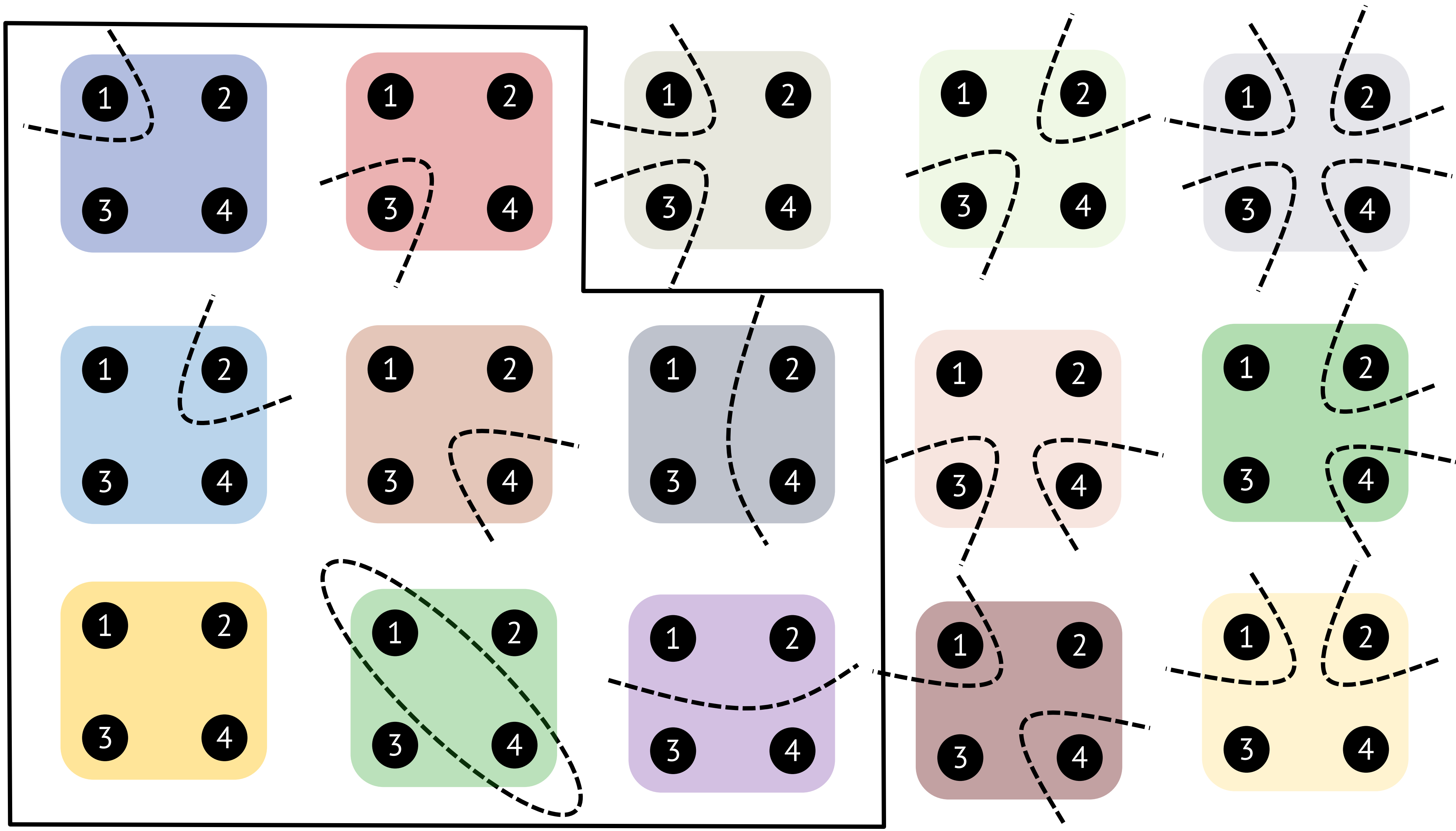


The hypergraph minimum s-t cut problem separates *s* and *t* in a way that minimizes the hypergraph cut.

This has a polynomial-time solution [Lawler 1973].

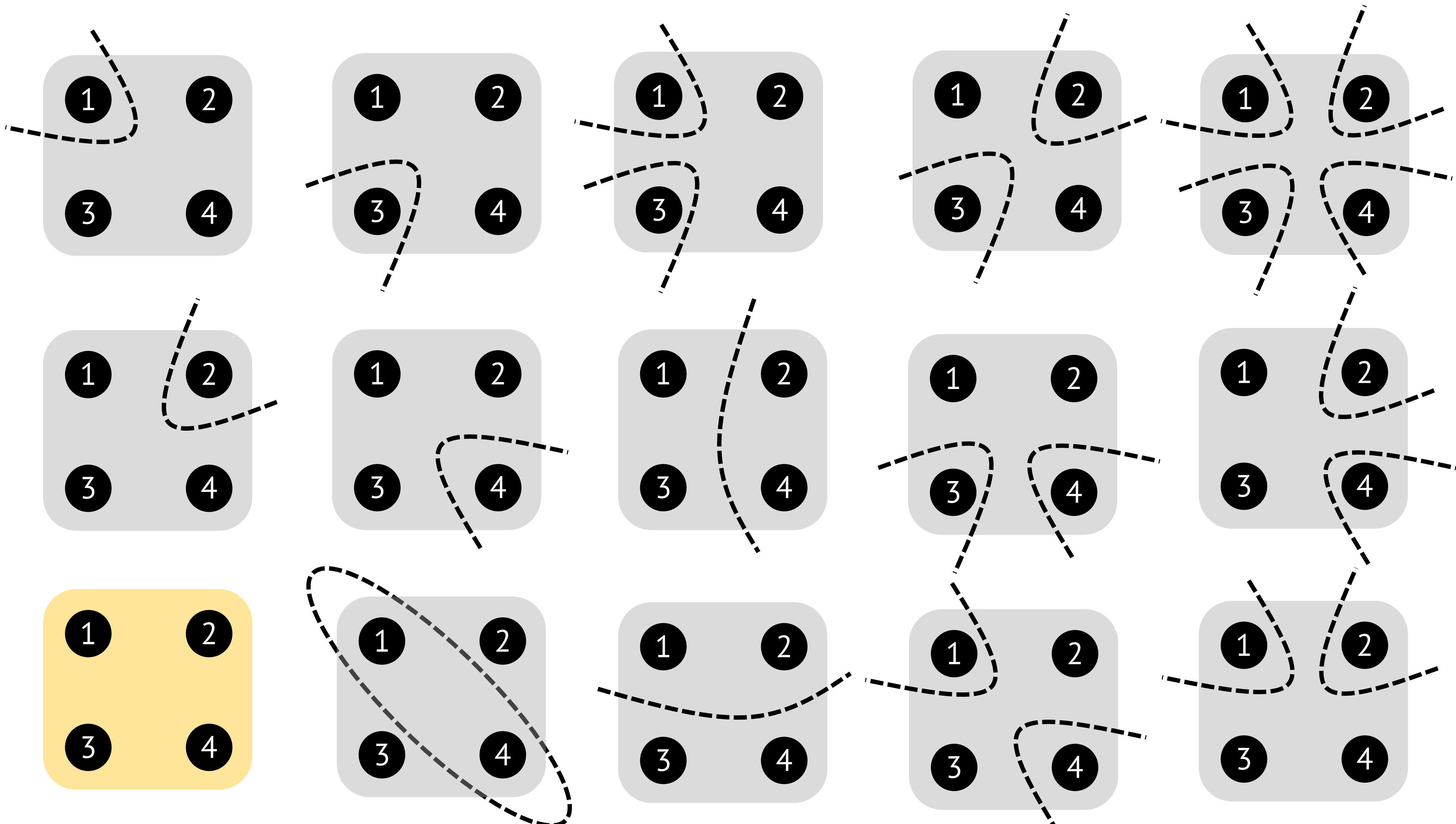
This cut function seems natural at first, but does it always make sense?

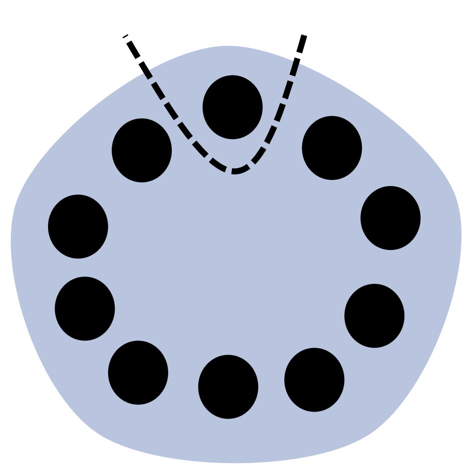
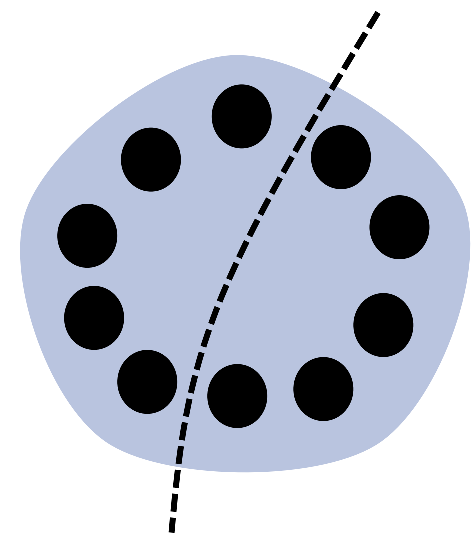
There are 14 distinct ways to cut a 4-node hyperedge



Seven when restricting to two clusters.

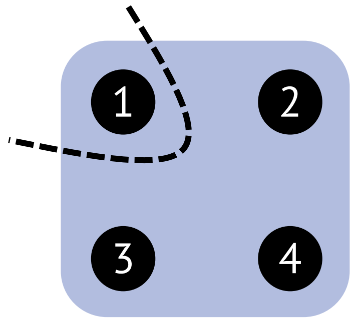
Here's how the standard hypergraph cut function sees them



Should we really treat  the same as  ?

We introduce a generalized class of hypergraph cut functions based on splitting functions.

A splitting function associates a penalty to each configuration of nodes in a hyperedge.



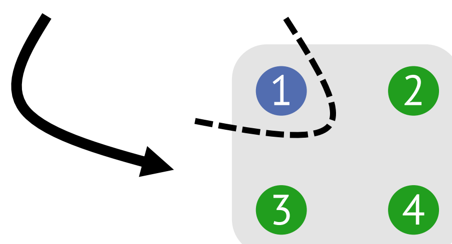
$\mathbf{w}_e(\{1\})$ = penalty for {1} vs. {2,3,4} split

Assumptions.

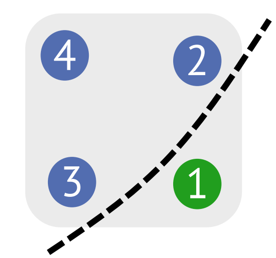
Uncut ignoring $\mathbf{w}_e(e) = \mathbf{w}_e(\emptyset) = 0$

Non-negativity $\mathbf{w}_e(U) \geq 0$ for all $U \subset e$.

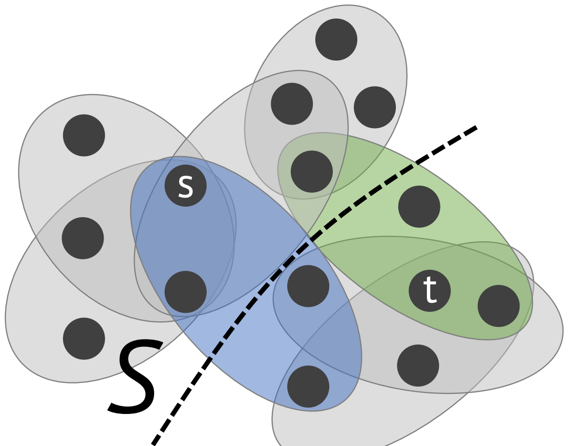
Symmetry $\mathbf{w}_e(U) = \mathbf{w}_e(e \backslash U)$ for all $U \subset e$.



$\mathbf{w}_e(\{1\}) = \mathbf{w}_e(\{2,3,4\})$



We focus on a very natural class of splitting functions.



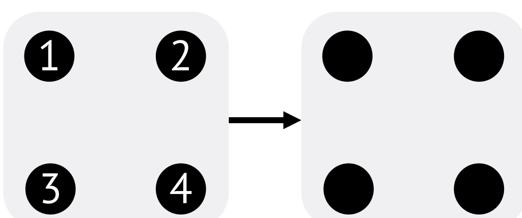
$\text{cut}_{\mathcal{H}}(S) = f(2) + f(1)$

Hypergraph minimum s-t cut problem.

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

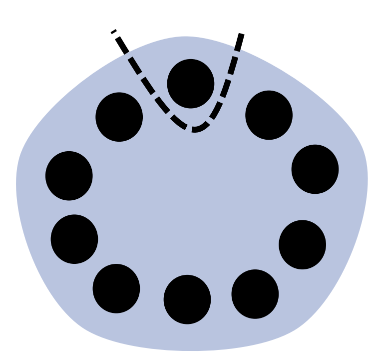
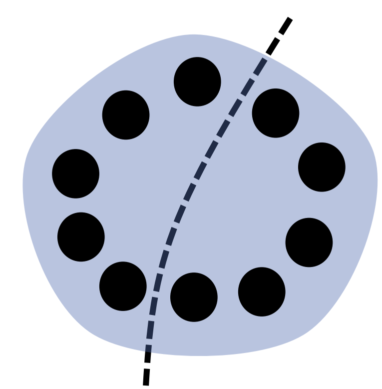
Assume all hyperedges of the same size have the same splitting function.

In theory, we could assign a completely different function to each hyperedge.



Cardinality-based splitting functions.

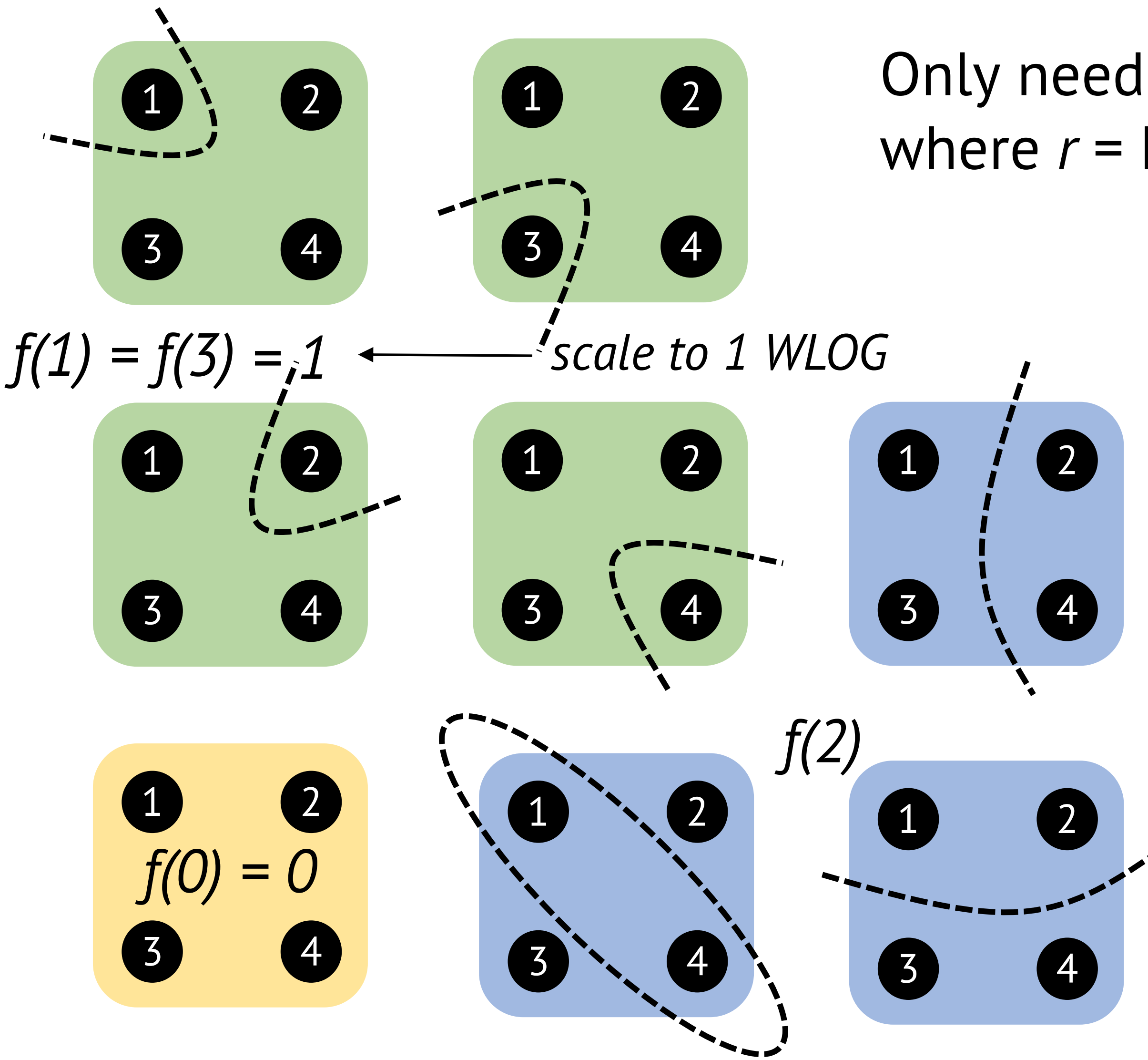
$$\mathbf{w}_e(U) = f(\min(|U|, |U \backslash e|))$$

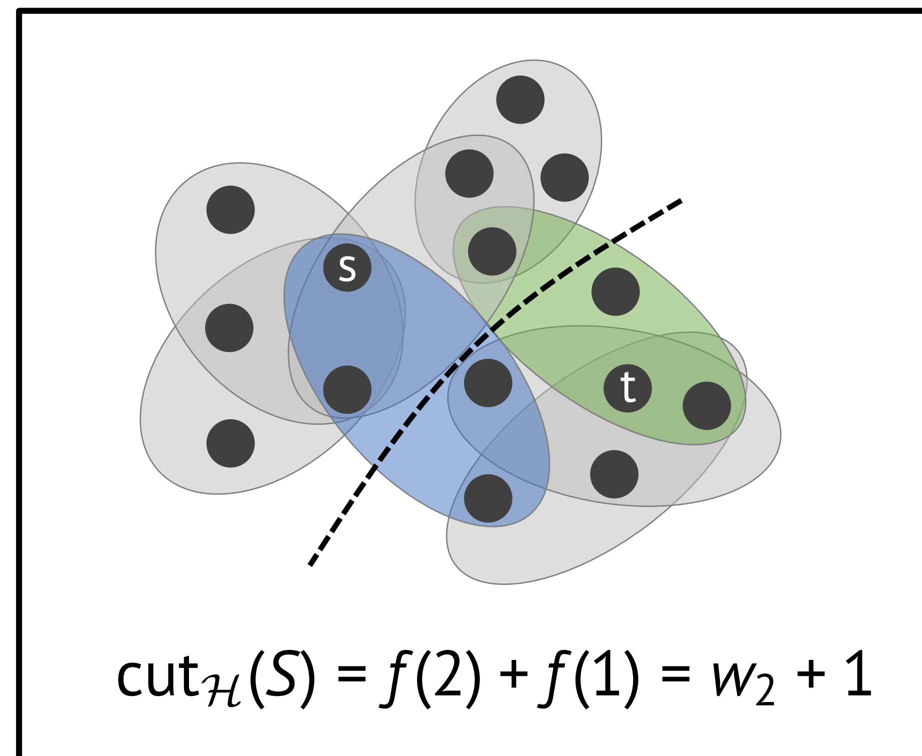
This allows us to distinguish between  and  .

Cardinality-based functions are easy to specify.

Only need to specify $f(1), f(2), \ldots, f(\lfloor r/2 \rfloor)$ where r = hyperedge size.

$$f(j) = w_j$$

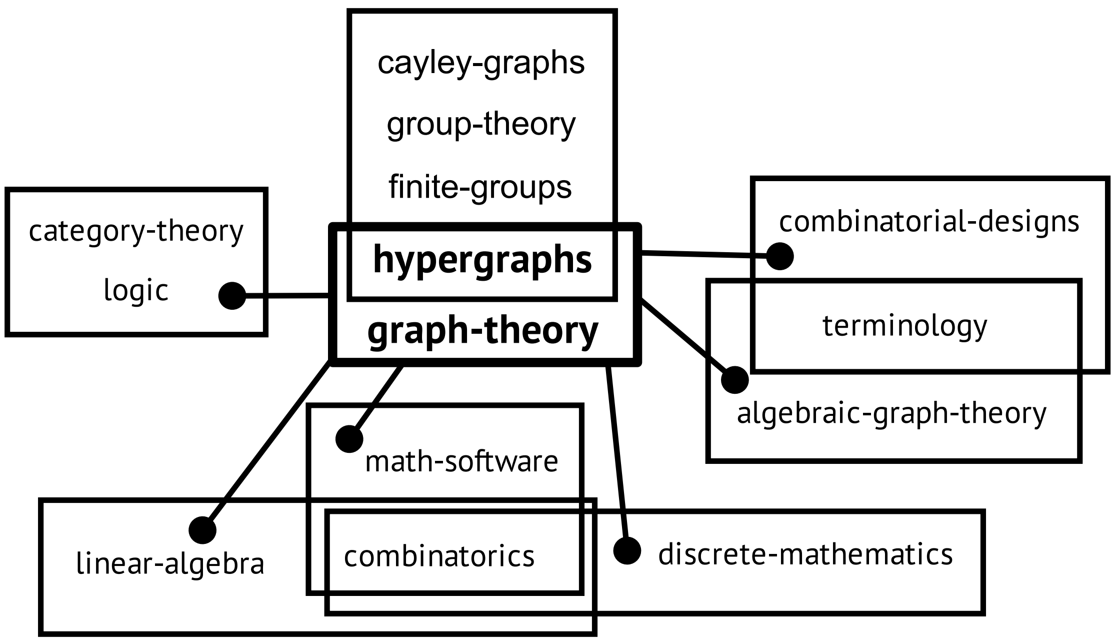




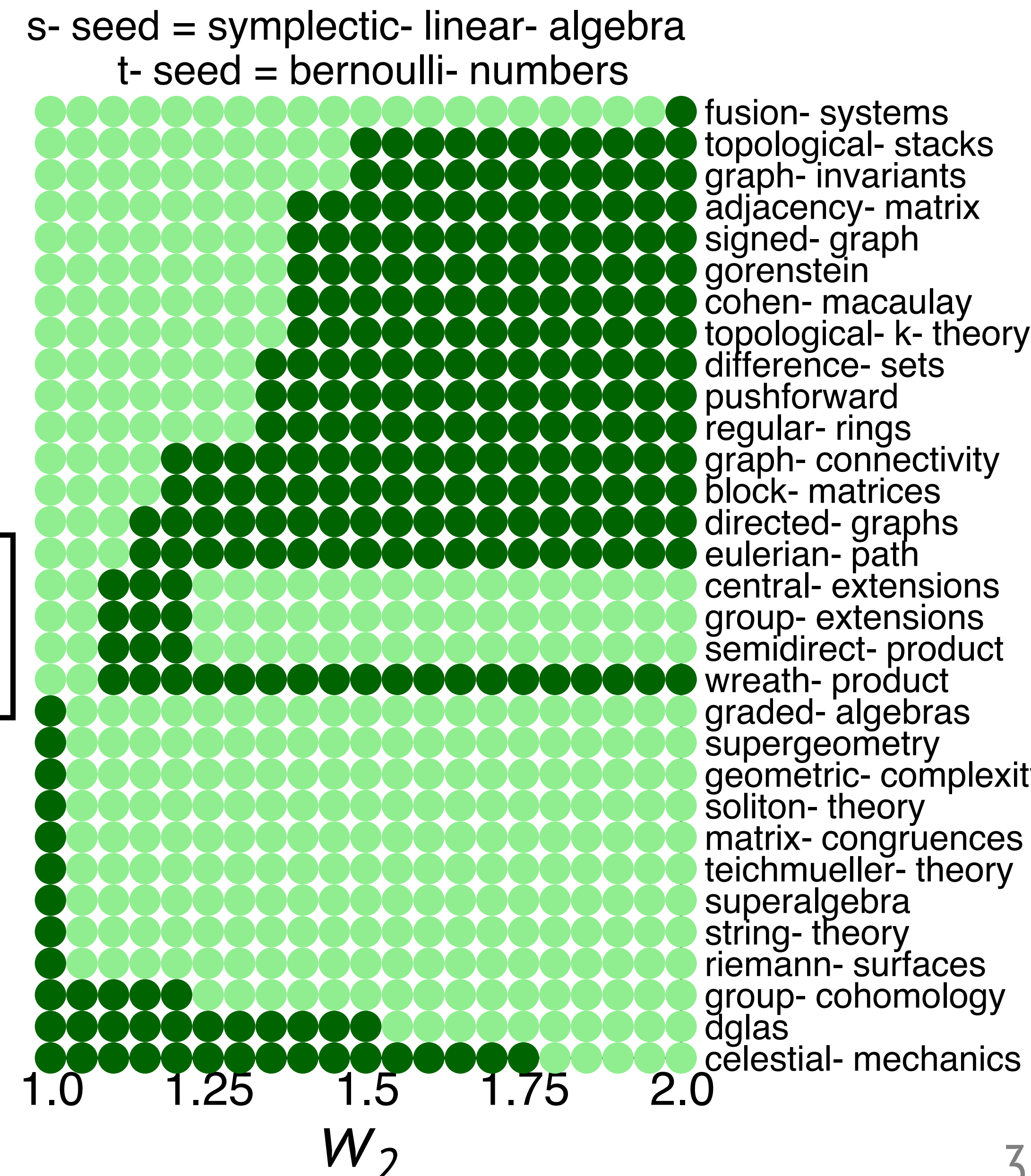
Different weights lead to different min cuts in practice.

node is on **s** side

node is on **t** side



node = tag on math.stackexchange.com
hyperedge = set of tags in same post

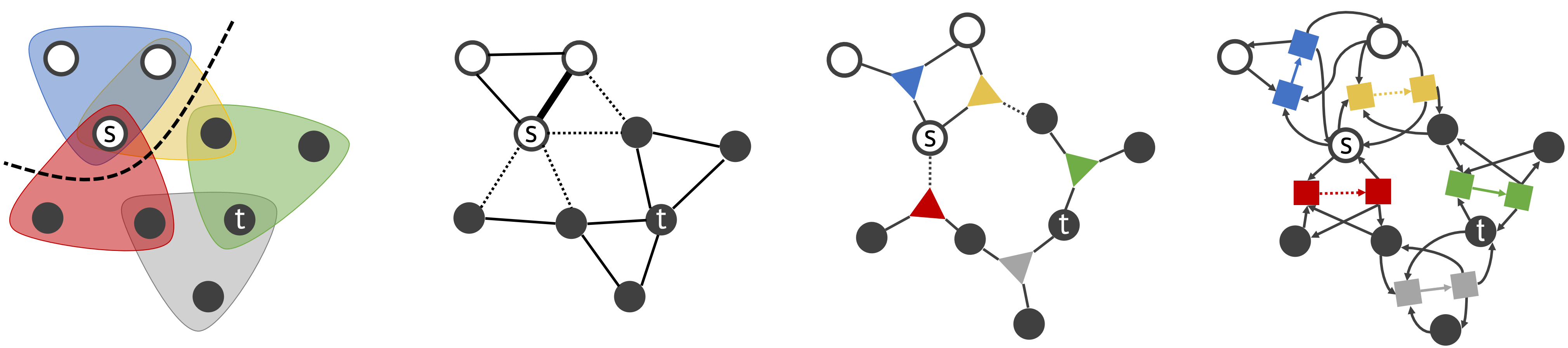


We solve hypergraph cut problems with graph reductions

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

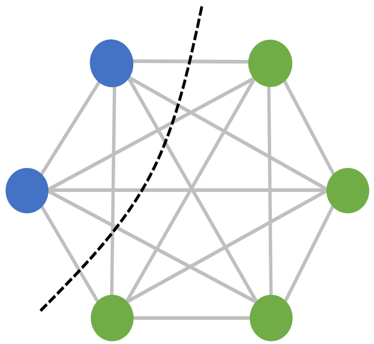
hyperedge | clique expansion | star expansion | Lawler gadget [1973]

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



...and then exactly solve the resulting graph s-t cut problem.

Existing gadgets model cardinality-based splitting functions.



Clique Gadget

Does not require adding new vertices

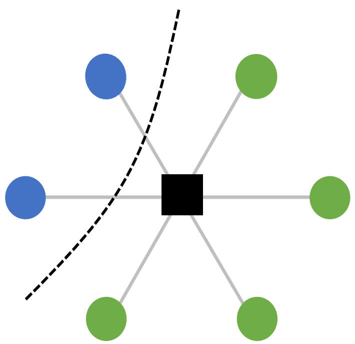
[Agarwal+ 06; Zhou+ 06; Benson+ 16]

models →

Quadratic penalty

$$\mathbf{w}_e(U) = |U| \cdot |e \setminus U|$$

i.e., $w_i = i \cdot (r-i)$



Star Gadget

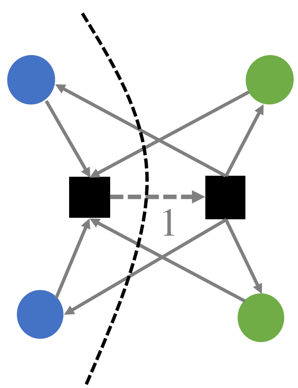
Equivalent to bipartite expansion of a hypergraph

[Hu-Moerder 85; Heuer+ 18]

models →

Linear penalty

$$\mathbf{w}_e(U) = \min\{|U|, |e \setminus U|\}$$



Lawler Gadget

Models the standard hypergraph cut function

[Lawler 73; Ihler+ 93; Yin+ 17]

models →

All-or-nothing penalty

$$\mathbf{w}_e(U) = \begin{cases} 1 & \text{if } U \in \{e, \emptyset\} \\ 0 & \text{otherwise} \end{cases}$$

How can we model other cardinality-based splitting functions?

Other cardinality-based functions are also used in other hypergraph clustering applications.

Used for consensus clustering.

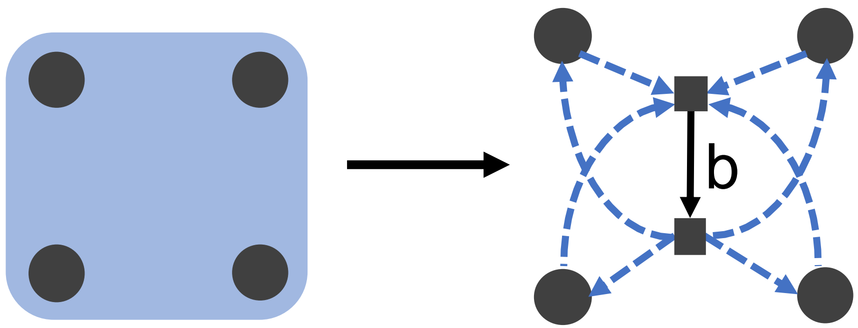
Discount cut $\mathbf{w}_e(U) = \min\{|U|^\alpha, |e \backslash U|^\alpha\}$ [Yaros-Imielinski 13]

L-M submodular $\mathbf{w}_e(U) = \frac{1}{2} + \frac{1}{2} \cdot \min\left\{1, \frac{|U|}{\lfloor \alpha |e| \rfloor}, \frac{|e \backslash U|}{\lfloor \alpha |e| \rfloor}\right\}$ [Li-Milenkovic 18]

Used for hypergraph spectral clustering.

No graph reduction strategy has been designed for these. Can we develop one?

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



C-B $\mathbf{w}_e(U) = f(\min(|U|, |e \backslash U|))$.
This gadget models $\min(|U|, |e \backslash U|, b)$.

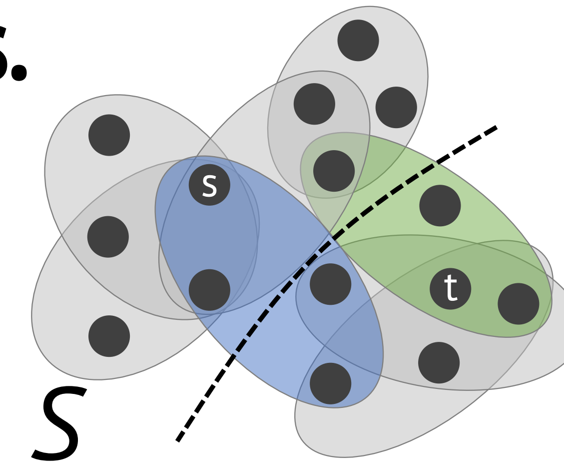
Theorem [Veldt-Benson-Kleinberg 20a]. Nonnegative linear combinations of the C-B gadget can model any submodular cardinality-based splitting function.

(F is submodular on X if $F(A \cap B) + F(A \cup B) \leq F(A) + F(B)$ for any $A, B \subseteq X$.)

All the data mining / machine learning applications of hypergraph cuts map to a submodular cardinality-based splitting function.

See also Graph Cuts for Minimizing Robust Higher Order Potentials, Kohli et al., 2008.

Submodularity is key to efficient algorithms.



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

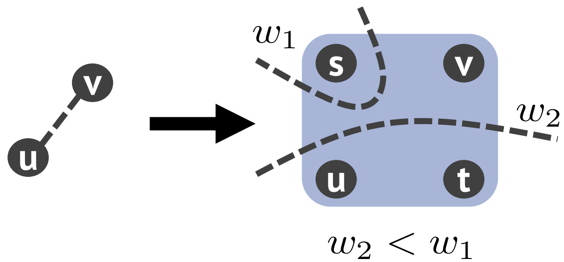
Cardinality-based splitting functions.

$$\mathbf{w}_e(U) = f(\min(|U|, |e \backslash U|))$$

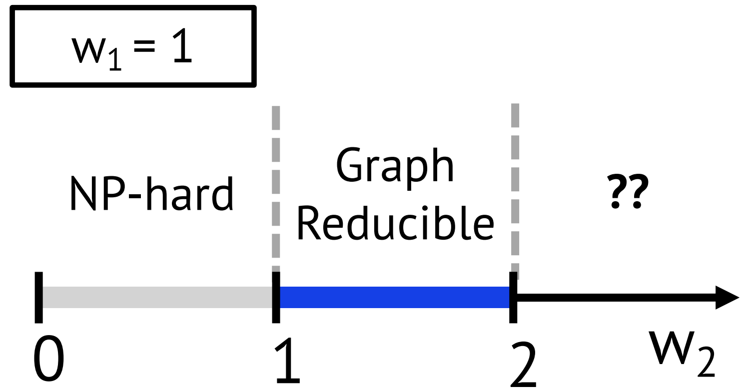
Theorem [Veldt-Benson-Kleinberg 20a]. The hypergraph min s-t cut problem with a cardinality-based splitting function is graph-reducible (via gadgets) *if and only if* the splitting function is submodular.

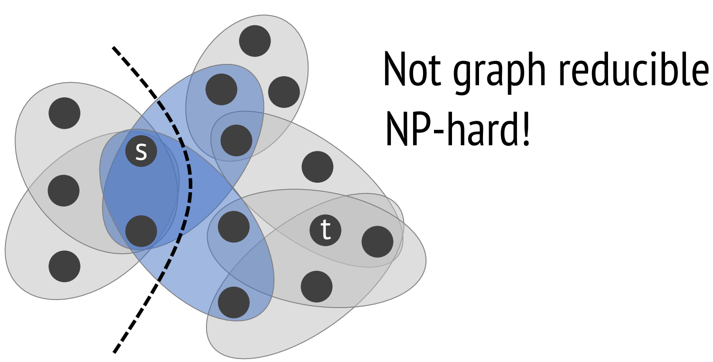
What happens when the splitting function isn't submodular?
Can we use some other algorithm?

Hardness and open questions for 4-node case.

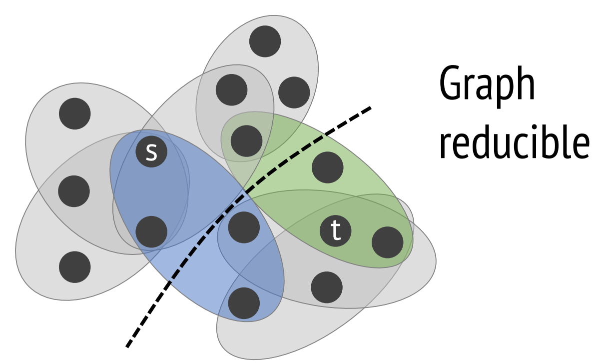


Reduction from max cut

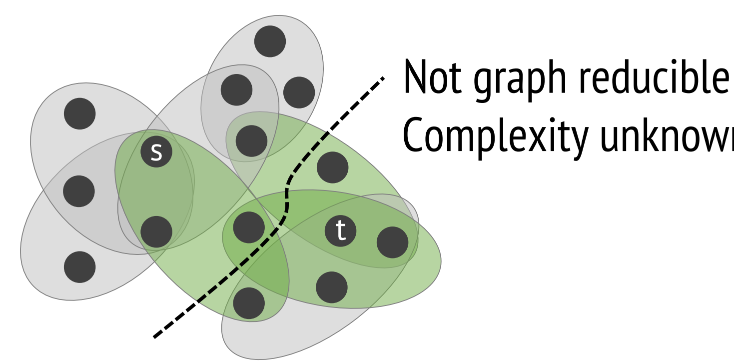




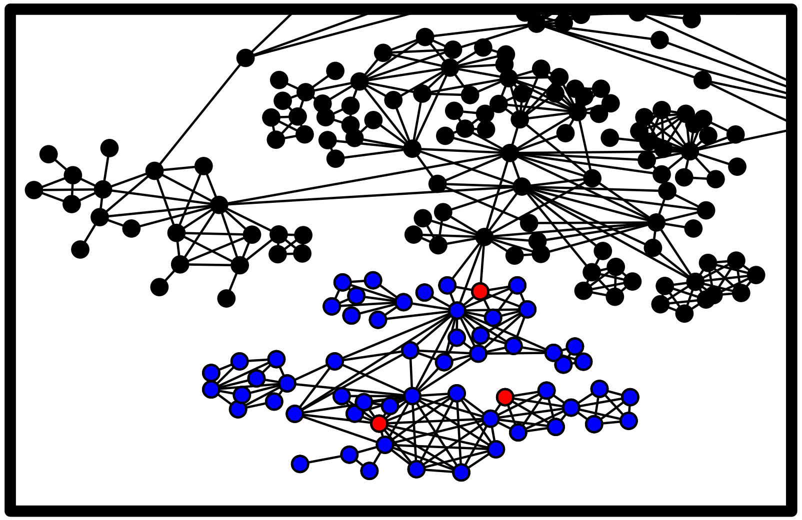
w_2 = 0.5 solution



w_2 = 1.5 solution



w_2 = 2.5 solution



The goal of **local graph clustering** is to find a good cluster ***S*** near a seed set ***R.***

Local Hypergraph Clustering

Minimizing Localized Ratio Cut Objectives in Hypergraphs

Veldt, Benson, Kleinberg **KDD 2020**

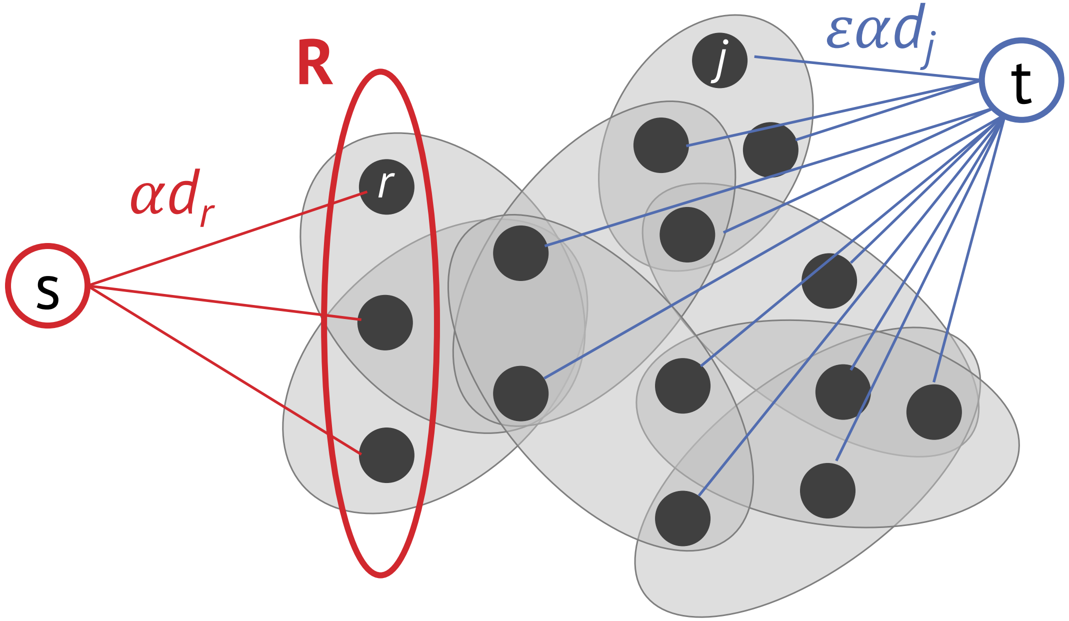
Examples.

Localize left atrial cavity in full body MRI [Veldt+ 19].

Finding a specific person's social communities [Fountoulakis+ 20].

Determine related products from co-purchasing data.

HyperLocal does localized hypergraph clustering by repeated hypergraph s-t cuts.



We introduce a new Hypergraph Local Conductance objective.

Hypergraph cut function

$$\mathbf{HLC}_{R,\varepsilon}(S) = \frac{\mathrm{cut}_{\mathcal{H}}(S)}{\mathrm{vol}_{\mathcal{H}}(S \cap R) - \varepsilon \mathrm{vol}_{\mathcal{H}}(S \cap \bar{R})}$$

Encourage overlap with seed set.

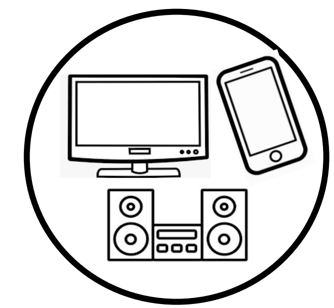
Discourage overlap outside seed set

Theorem [Veldt-Benson-Kleinberg 2020b]
If $\mathrm{cut}_{\mathcal{H}}(S)$ is any cardinality-based submodular hypergraph cut function, the HLC objective can be minimized in polynomial time by solving a bounded number of hypergraph minimum s-t cut problems.

Detecting Amazon product categories from review data

Runtime and accuracy for detecting products of the same category from seed nodes

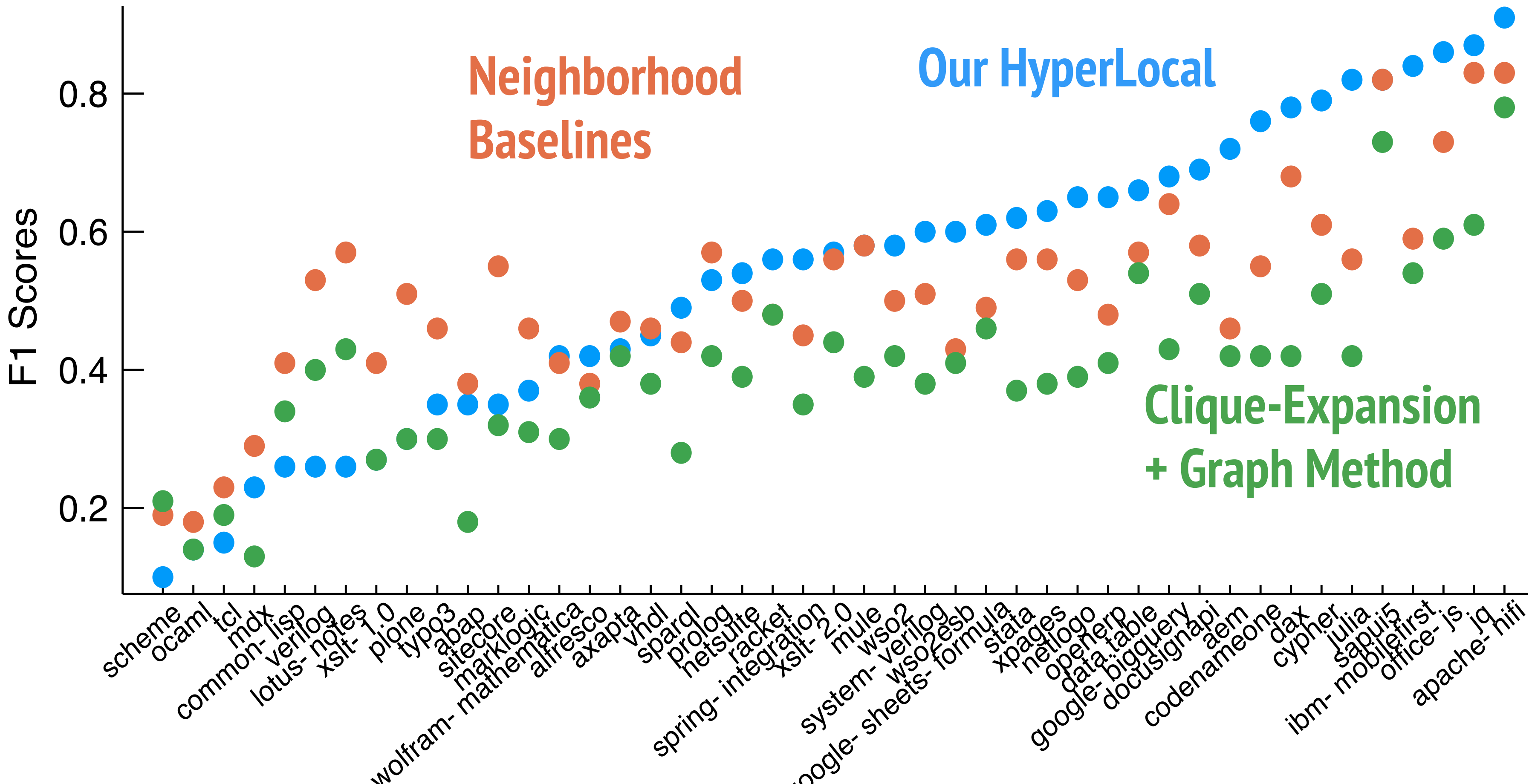
Cluster	$\lvert T \rvert$	time (s)	HyperLocal	Baseline1	Baseline2
Amazon Fashion	31	3.5	**0.83**	0.77	0.6
All Beauty	85	30.8	**0.69**	0.60	0.28
Appliances	48	9.8	**0.82**	0.73	0.56
Gift Cards	148	6.5	**0.86**	0.75	0.71
Magazine Subscriptions	157	14.5	**0.87**	0.72	0.56
Luxury Beauty	1581	261	**0.33**	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
- Product categories provide cluster labels
- All-or-nothing penalty ($w_i = 1$).

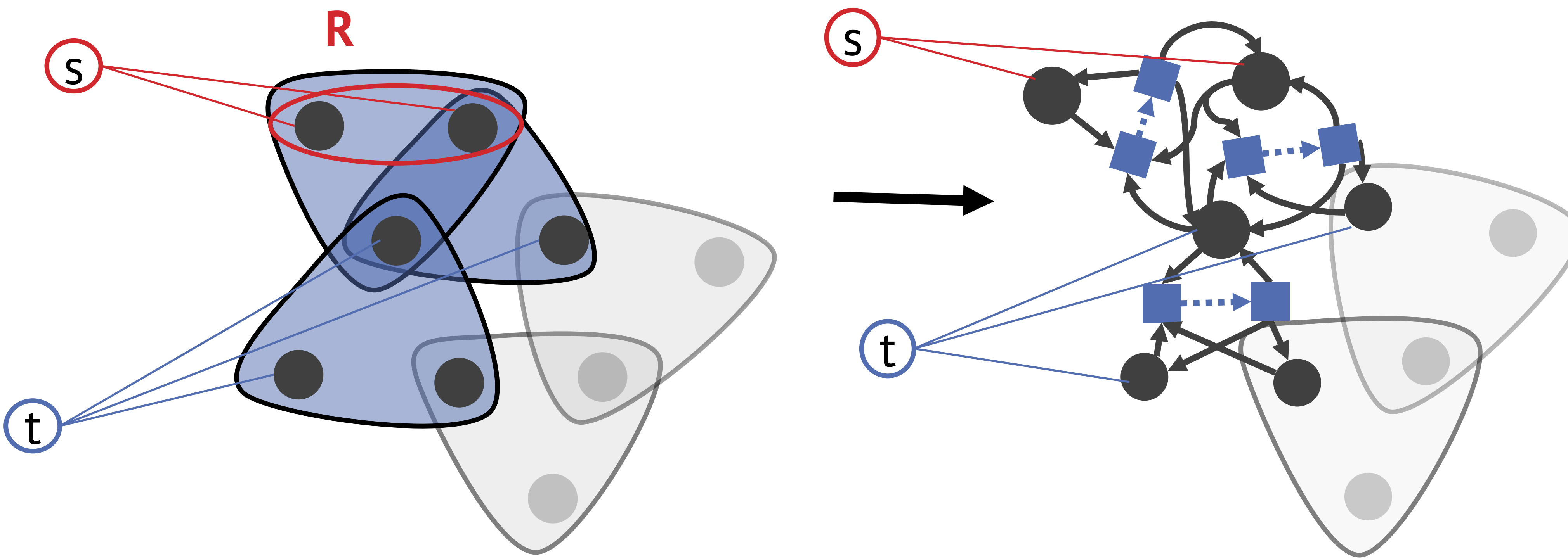
Max hyperedge size ~9.3k nodes!

Detecting online forum questions on the same topic



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

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



The resulting algorithm is very fast because it doesn't have to explore the entire hypergraph.

HyperLocal has strong theoretical guarantees.

Theorem [Veldt-Benson-Kleinberg 2020b]
Runtime guarantees.
The runtime of HyperLocal depends only on the size of the seed set **R**, not the size of the hypergraph.

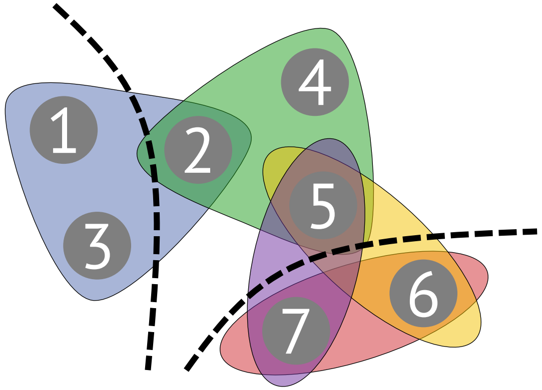
$$\mathbf{HLC}_{R,\varepsilon}(S) = \frac{\mathrm{cut}_{\mathcal{H}}(S)}{\mathrm{vol}_{\mathcal{H}}(S \cap R) - \varepsilon \mathrm{vol}_{\mathcal{H}}(S \cap \bar{R})}$$

Hypergraph localized conductance

Theorem [Veldt-Benson-Kleinberg 2020b]
Cluster detection guarantees.
If **R** overlaps enough with a target set **T**, HyperLocal will find a cluster **S** almost as good as **T**.

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

Hypergraph Normalized Cut



In global clustering, we
assign **every** node to **one** cluster.

Global Clustering and Community Detection

Generative Hypergraph Clustering:
from Blockmodels to Modularity
Chodrow, Veldt, Benson **arXiv 2101.09611**

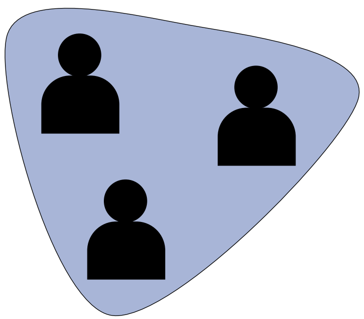
Examples.

Find social communities based on group interactions

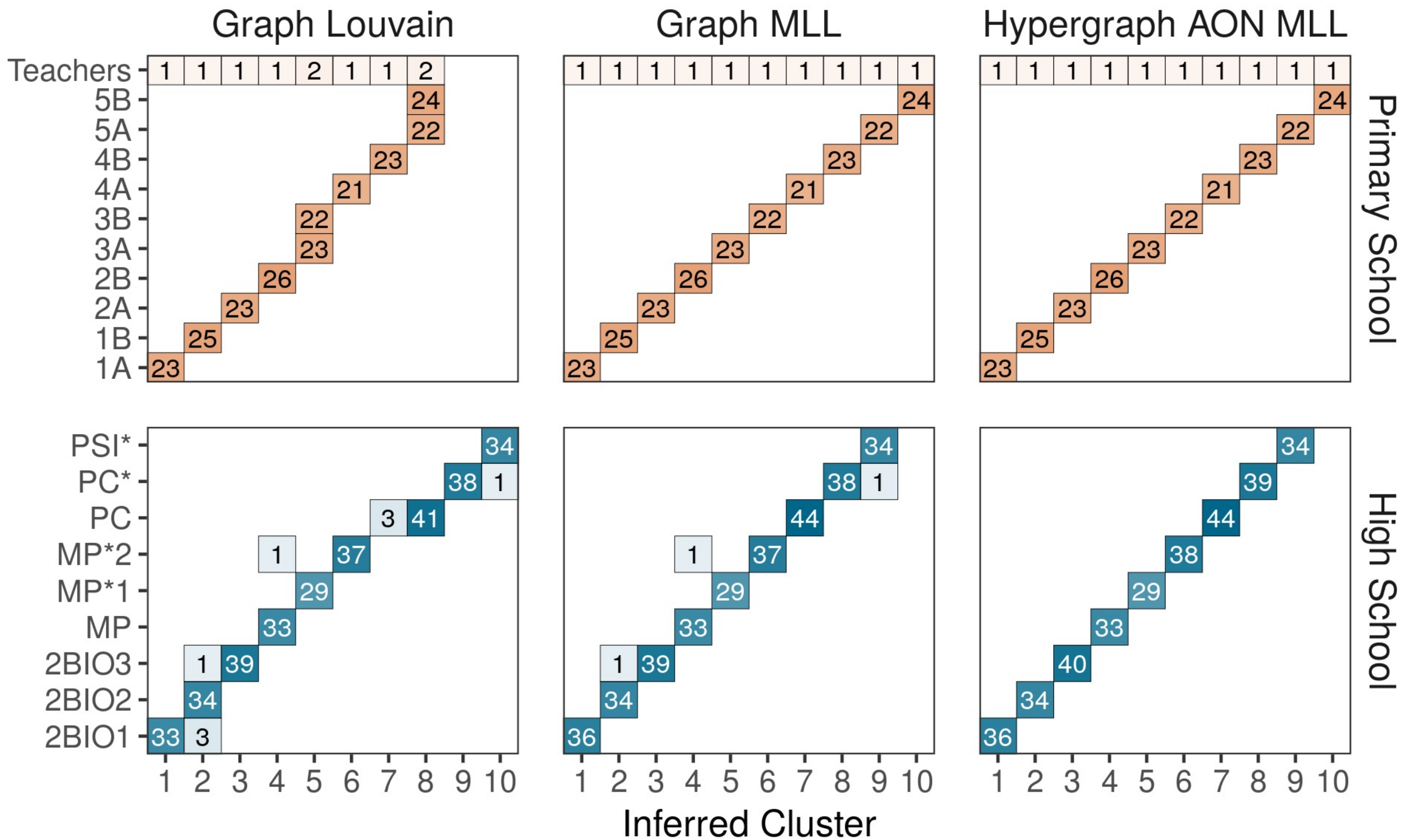
Separate retail products into different departments/categories

Minimizing communication costs in parallel computing [Çatalyürek-Aykanat 99; Kabiljo+ 17]

Contact hypergraphs



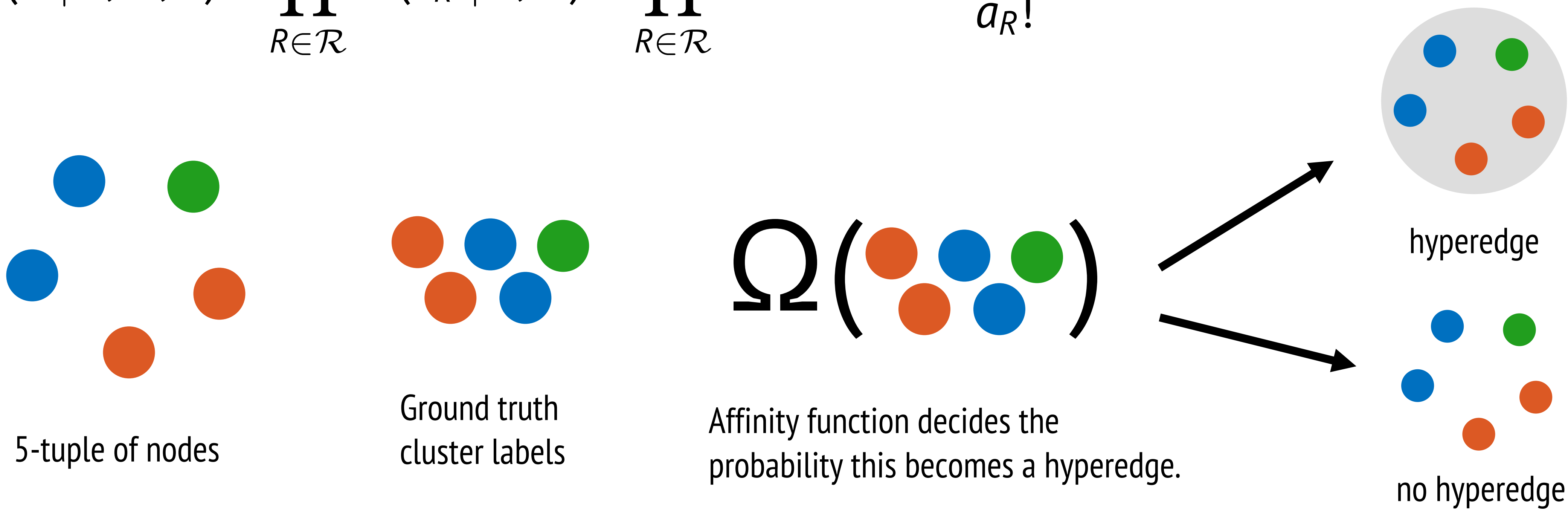
Hyperedge = group of students that interact during the day. Measured by wearable sensors



Our new hypergraph method beats graph methods at identifying groups of students (nodes) that belong to the same class (cluster) based on group interactions (hyperedges).

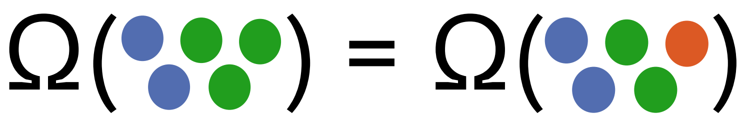
We develop a generative model for hypergraph clustering.

$$\Pr(A \mid z, \Omega, \theta) = \prod_{R \in \mathcal{R}} \Pr(a_R \mid z, \Omega) = \prod_{R \in \mathcal{R}} \frac{e^{-b_R \sigma(\theta_R)\Omega(z_R)} (b_R \sigma(\theta_R)\Omega(z_R))^{a_R}}{a_R!}$$



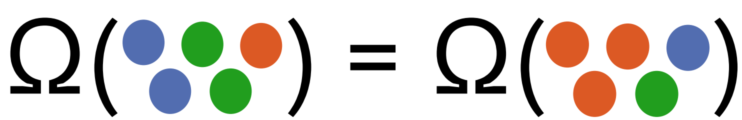
Maximum likelihood estimation to infer cluster labels and affinity functions involves minimizing hypergraph cuts.

Different affinity functions correspond to different types of hypergraph cut penalties

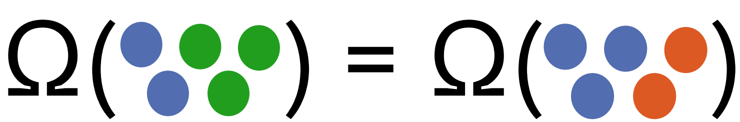


Example 1 (all-or-nothing). The affinity function depends only on whether all nodes are from the same ground truth cluster.

Optimizing the maximum likelihood function involves minimizing an AON hypergraph cut penalty.



Example 2 (Group Number). The affinity function depends only on the number of clusters appearing in a hyperedge.



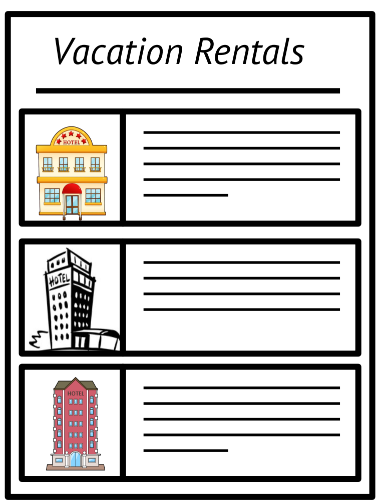
Example 3 (Relative Plurality). The affinity function depends only on the difference in count of top two most frequent clusters.

We can use true cluster labels to evaluate affinities.

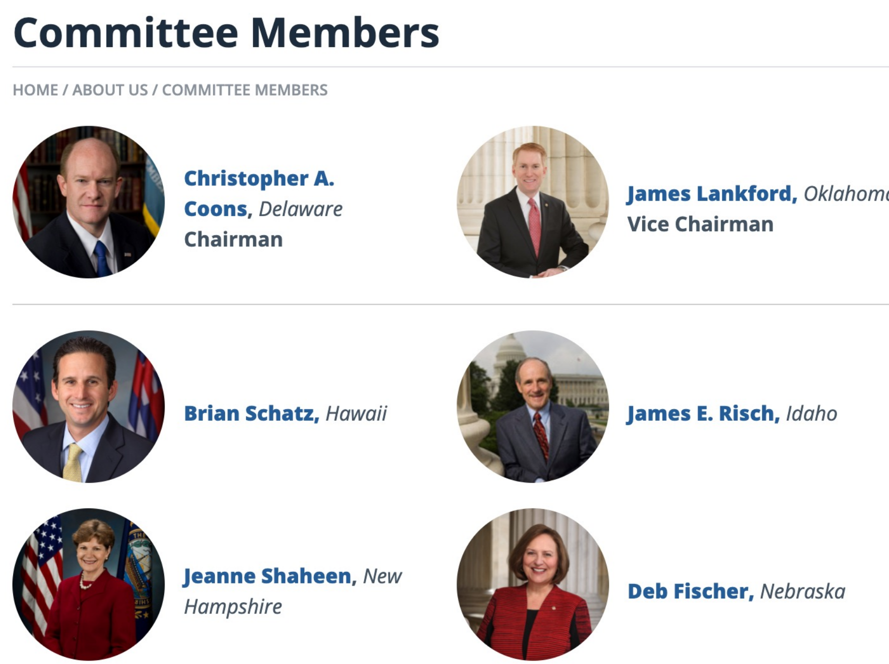
Bayesian information criterion (BIC)

	all-or-nothing	group number	relative plurality	
`trivago-clicks`	**1.6854**	1.6866	2.0257	$\times 10^8$
`house-committees`	2.7128	2.7128	**2.7119**	$\times 10^5$
`senate-committees`	9.7934	9.7934	**9.7736**	$\times 10^4$

Trivago hypergraph

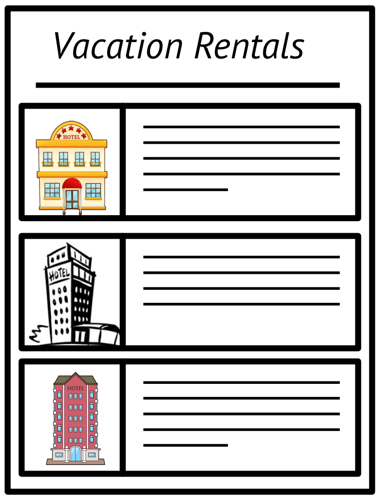


Hyperedge = set of vacation rentals that a user "clicks out" on during a browsing session

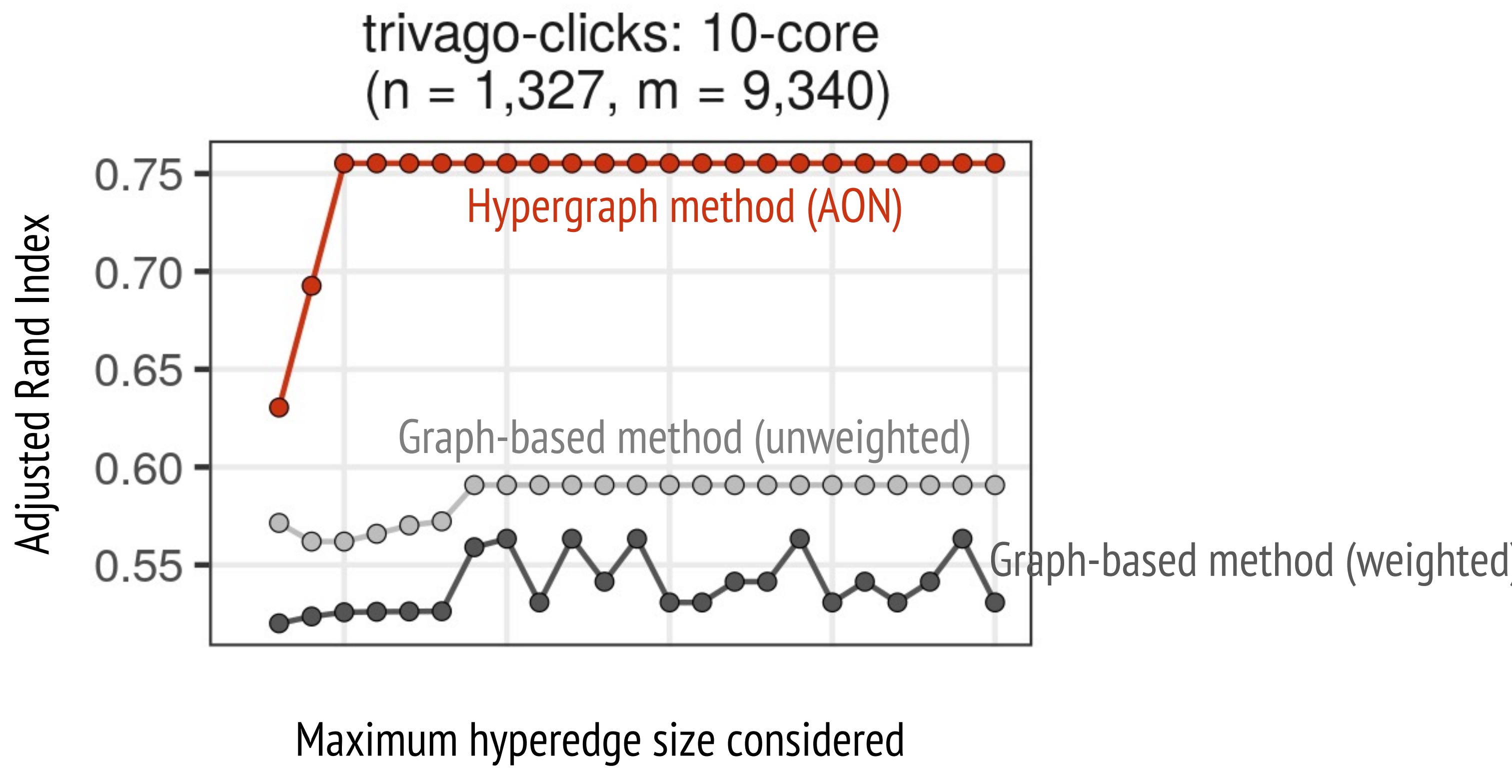


We can use max. likelihood estimation to infer clusters.

Trivago hypergraph



Hyperedge = set of vacation rentals that a user "clicks out" on during a browsing session

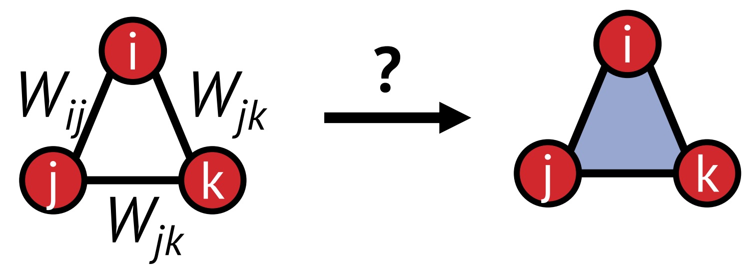


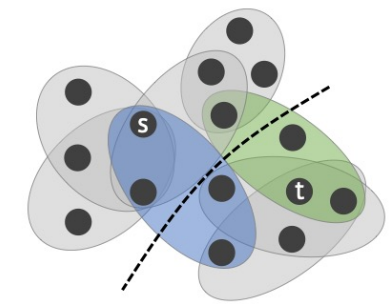
Our new hypergraph method beats graph methods at identifying vacation rentals (nodes) from the same country (cluster) based on browsing behavior.

Recap

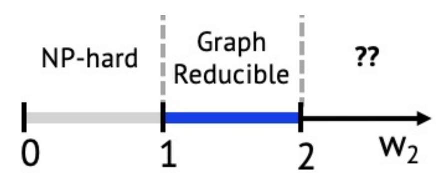
Hypergraphs model rich higher-order structure.

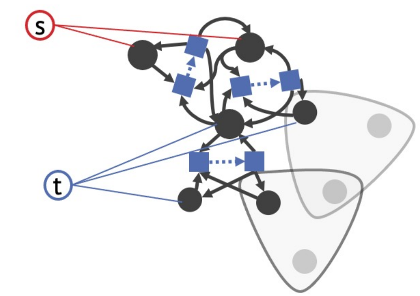
We can revisit classical problems and gain fresh insights.



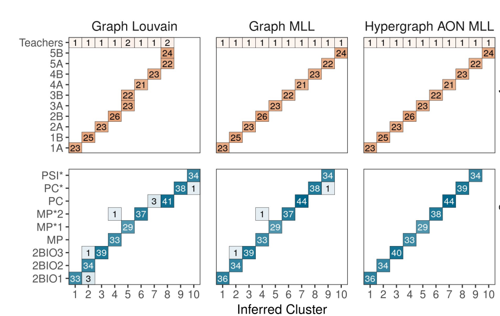


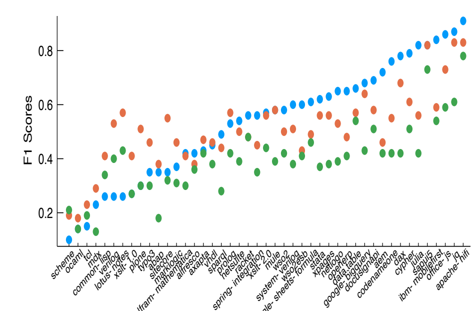
This are many new theoretical and algorithmic questions.





Applications to higher-order data analysis are abundant!





SIAM NEWS BLOG

Research | January 21, 2021

Print

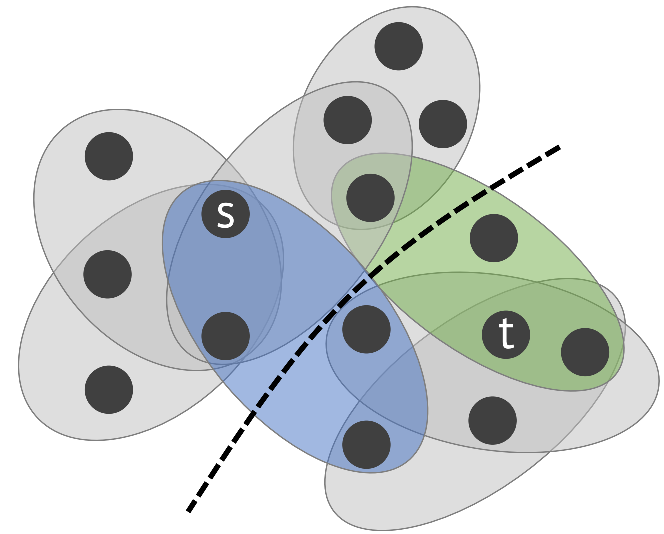
Higher-order Network Analysis Takes Off, Fueled by Old Ideas and New Data

By Austin R. Benson, David F. Gleich, and Desmond J. Higham

arXiv:2103.05031

Higher-order network science with hypergraph cuts

THANKS! Austin R. Benson
`http://cs.cornell.edu/~arb`
@austinbenson
arb@cs.cornell.edu



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

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