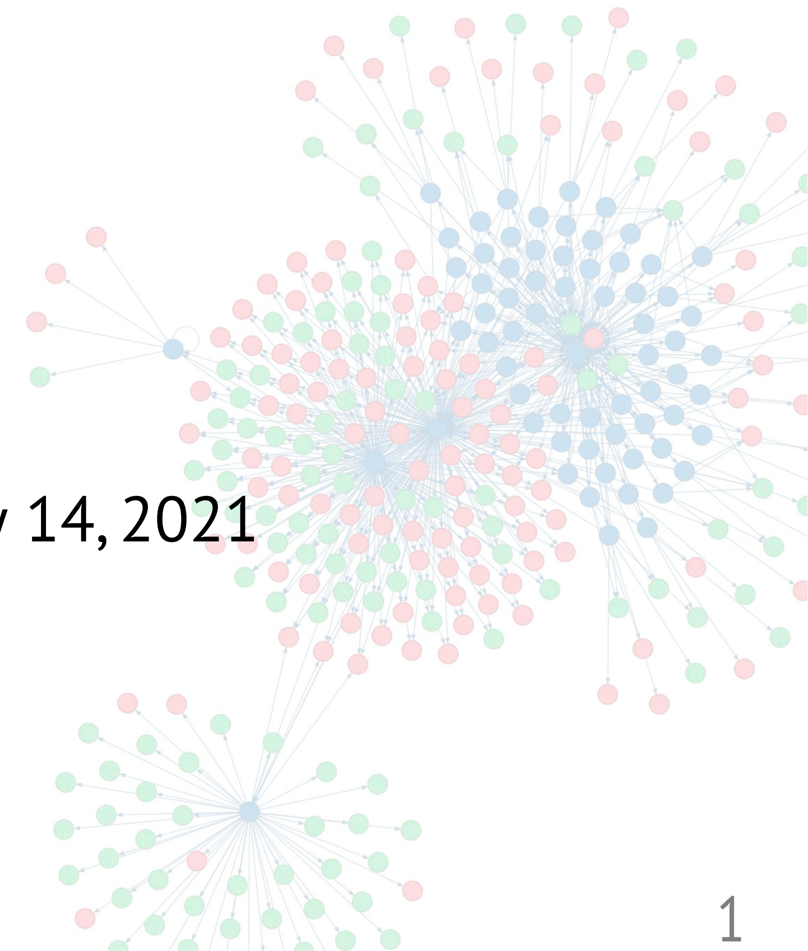
Hypergraph cuts with general splitting functions

Austin Benson · Cornell University

European Conference on Operational Research · July 14, 2021

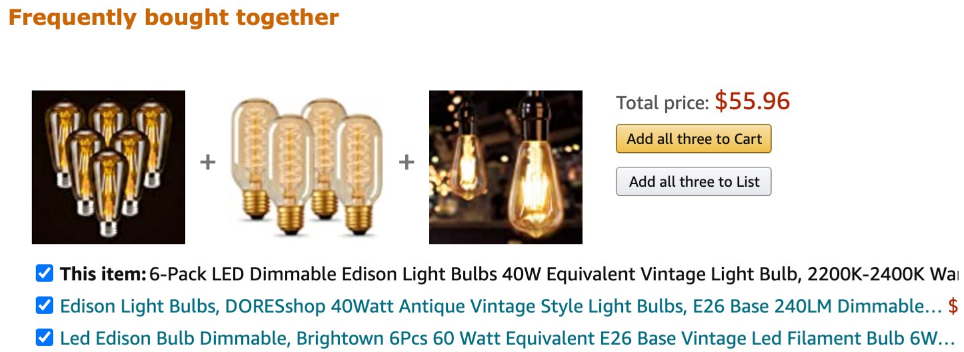


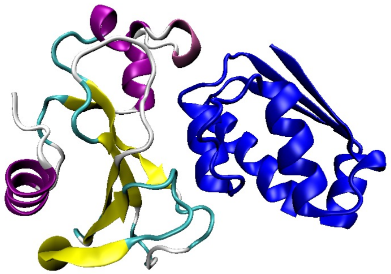
Graph or network data modeling important complex systems are everywhere.

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





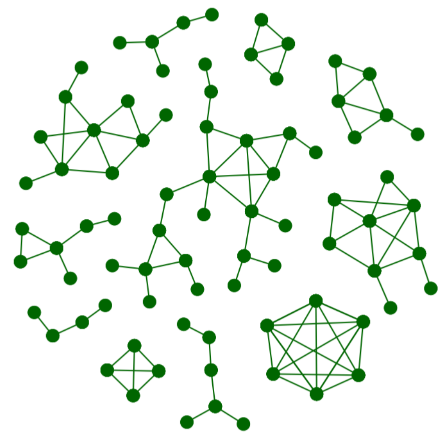
Cell biology
nodes are proteins
edge between two proteins that interact

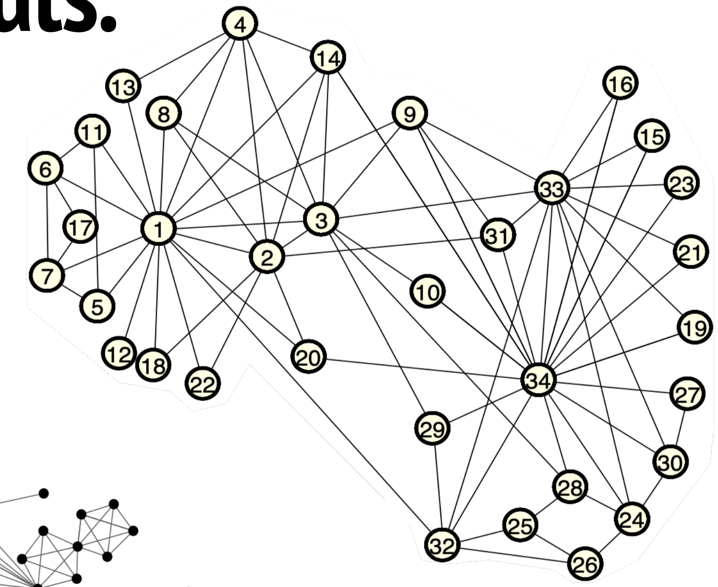
In network science, a wide array of applications rely on finding graph clusters and small graph cuts.

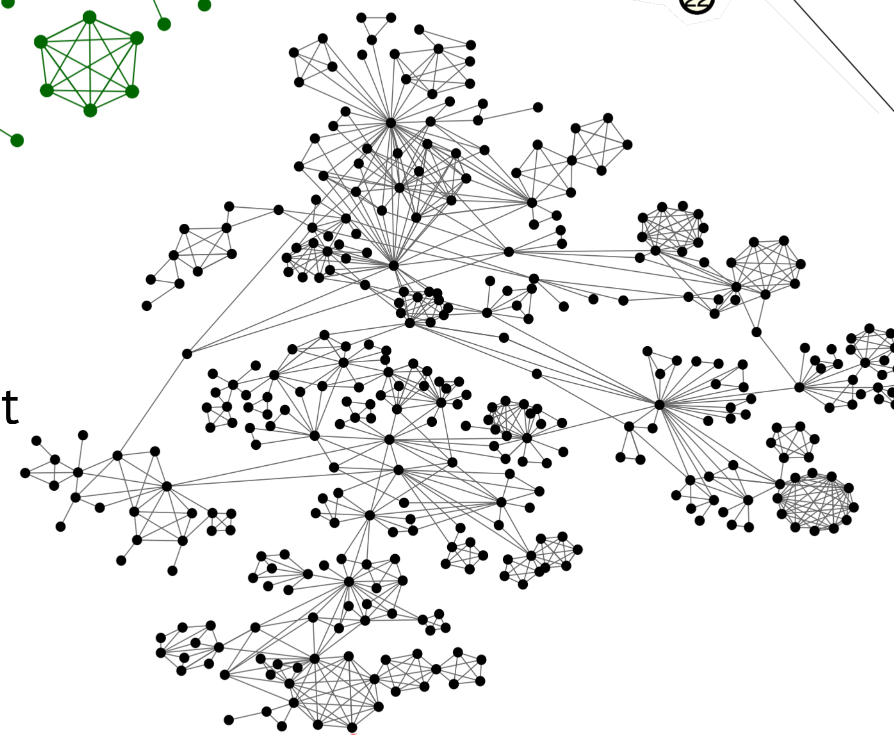
Applications

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

⋮



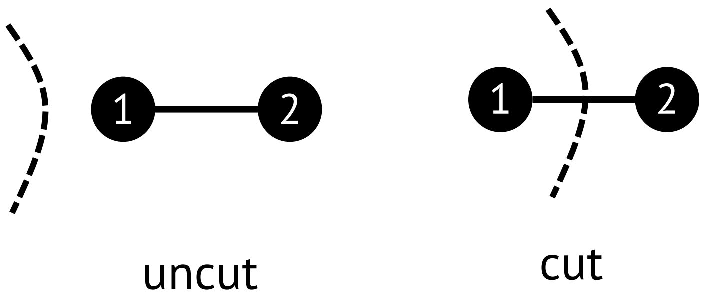




Cluster = densely connected node set that is sparsely connected to rest of graph

Cut = number of edges crossing a cluster boundary

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

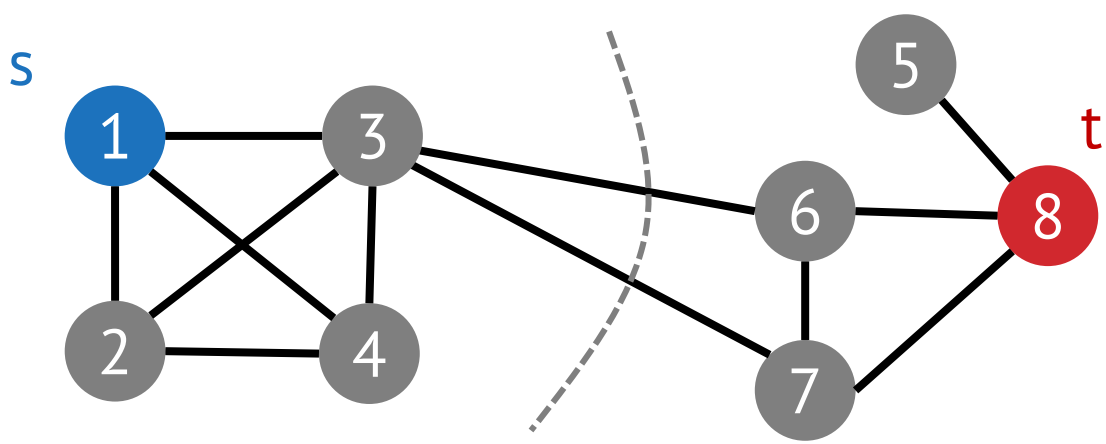


An edge is cut if its nodes are separated.

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



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

cut$(S) = 2$

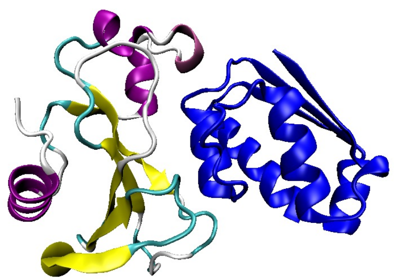
The penalty for cutting an edge is its weight.

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

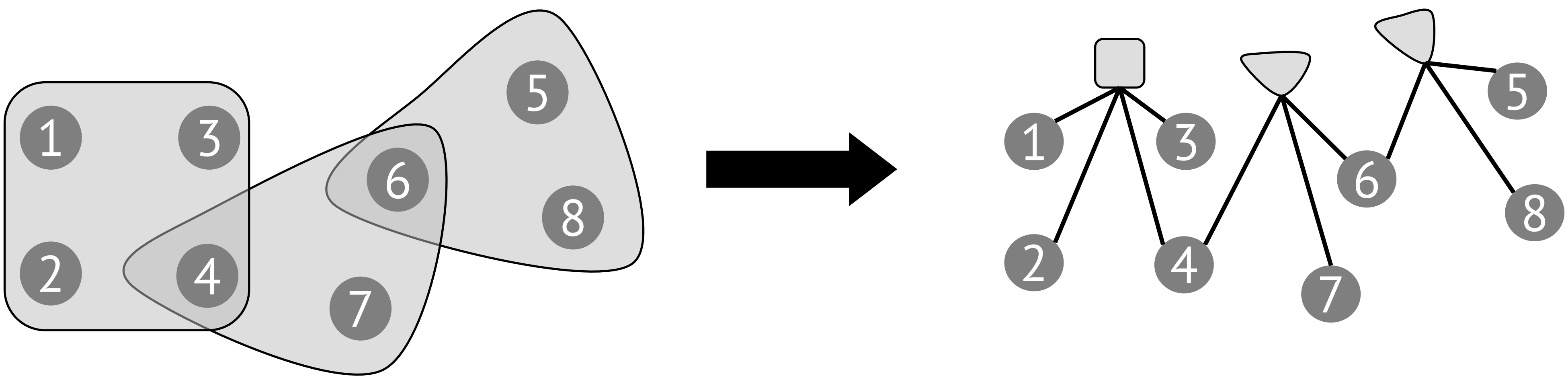
Commerce
nodes are products
several products can be purchased at once



Cell biology
nodes are proteins
protein complexes may involve several proteins

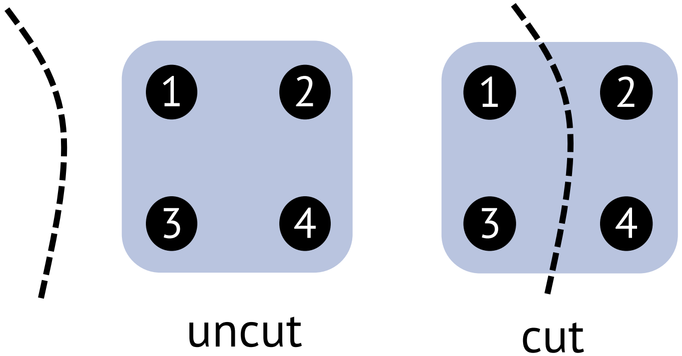
How do we define hypergraph cut problems?
Once defined, how do we solve them?

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



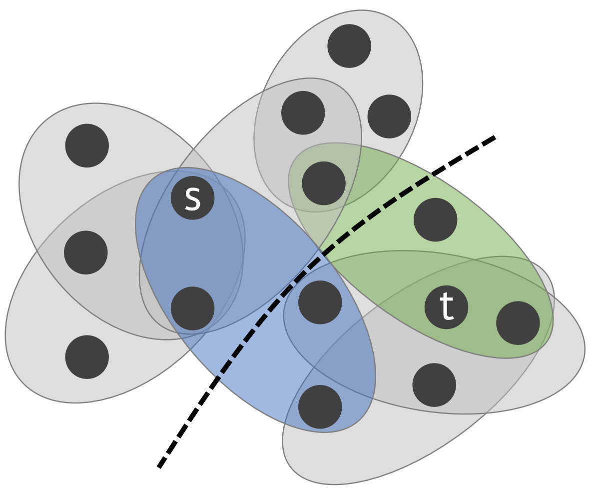
What exactly is this cut measuring? Is it right for applications? Are there alternatives?

~~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.

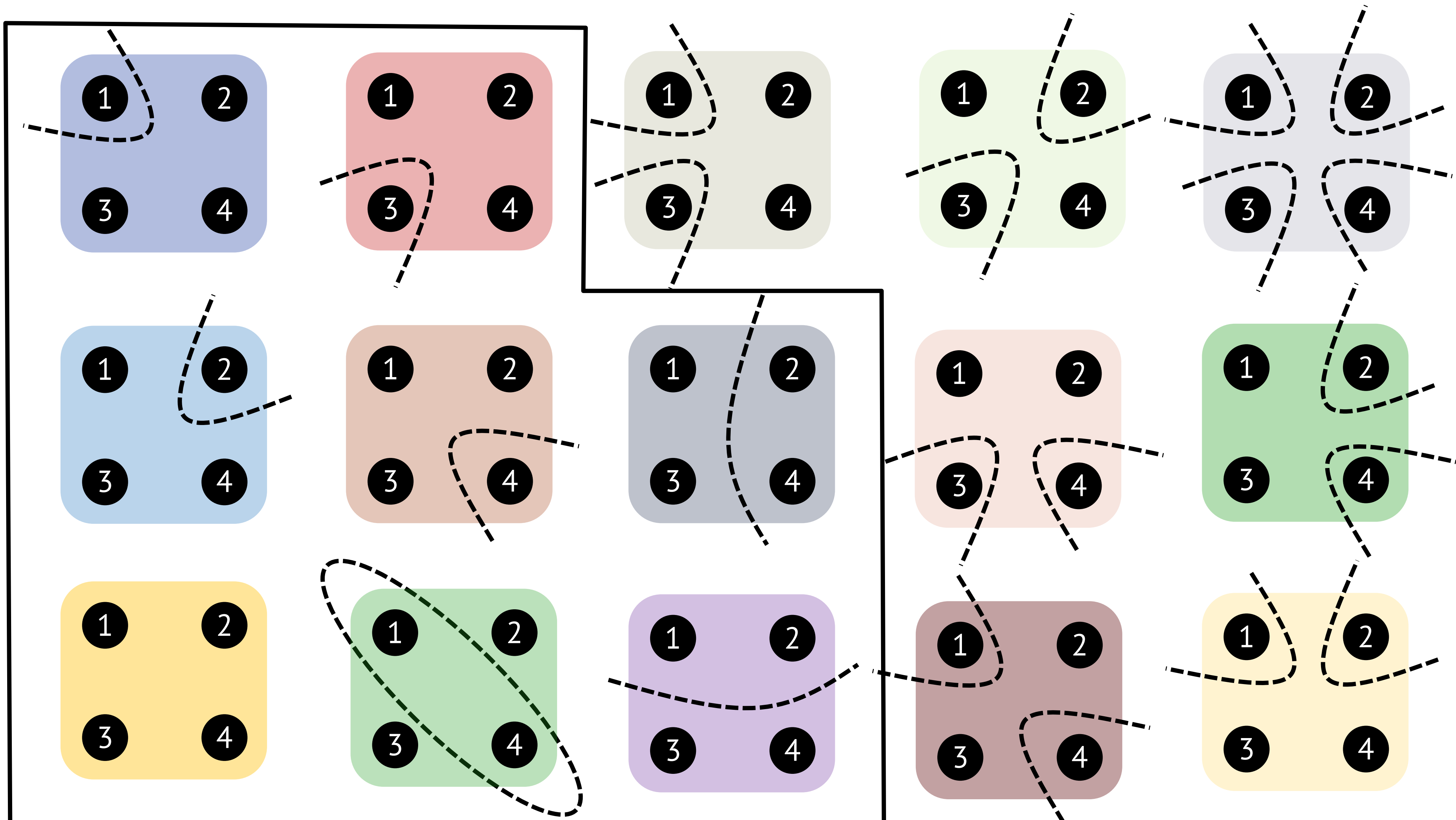


The hypergraph minimum s-t cut problem separates *s* and *t* in a way that minimizes the hypergraph cut.

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

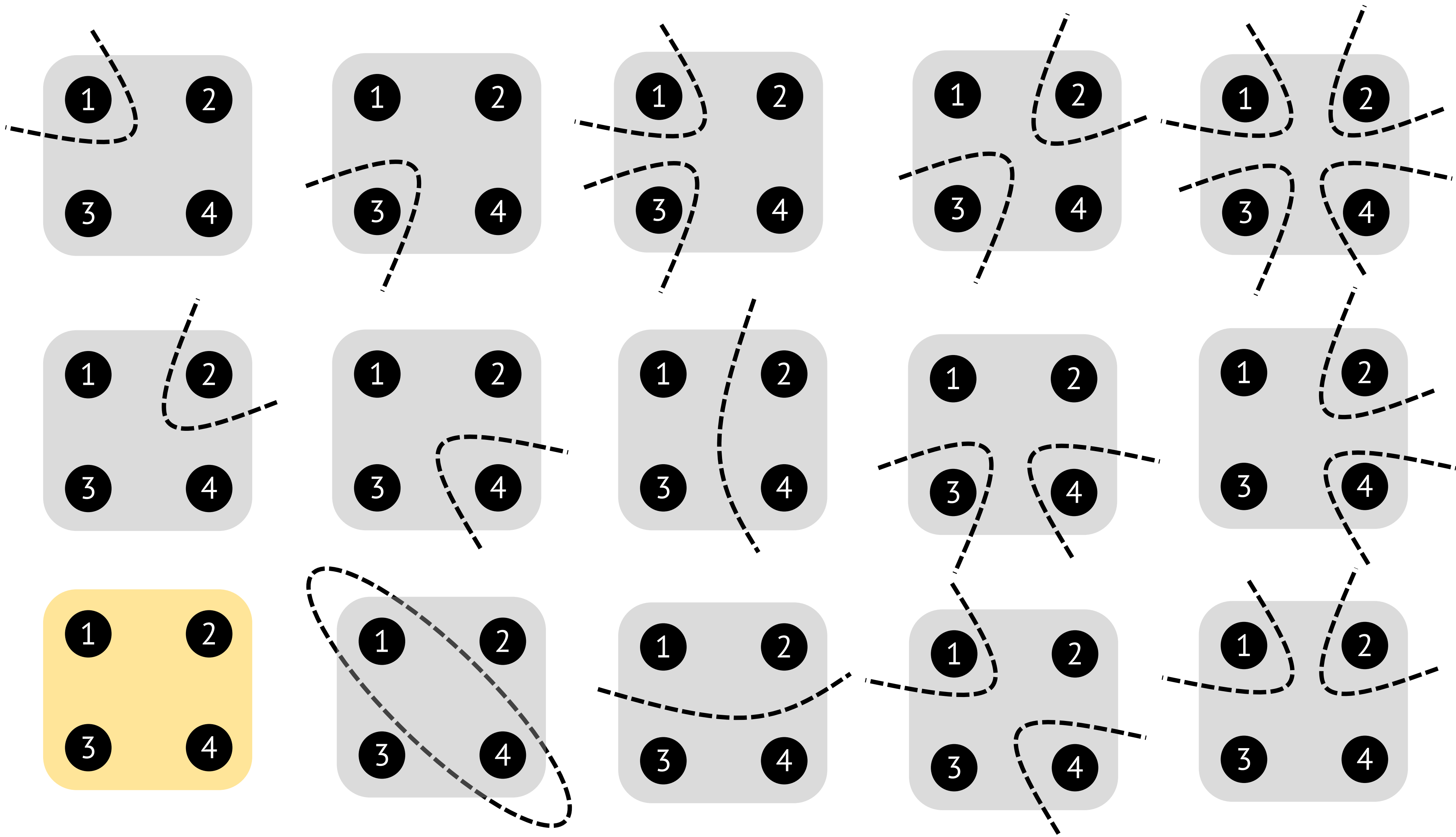
This cut function seems natural at first, but does it always make sense?

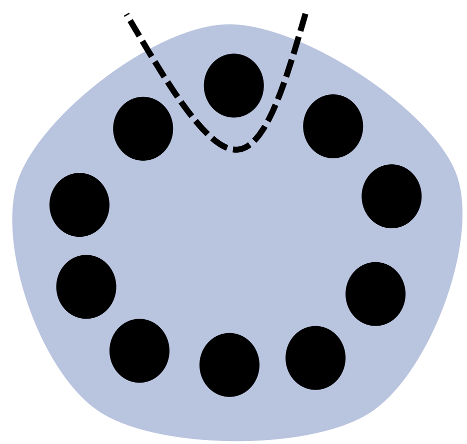
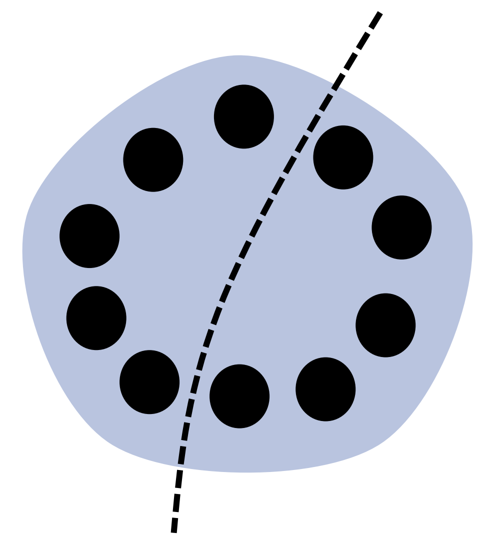
There are 14 distinct ways to cut a 4-node hyperedge



Seven when restricting to two clusters.

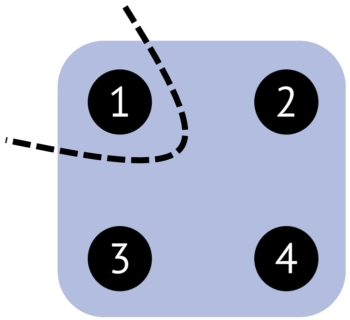
Here's how the standard hypergraph cut function sees them



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.



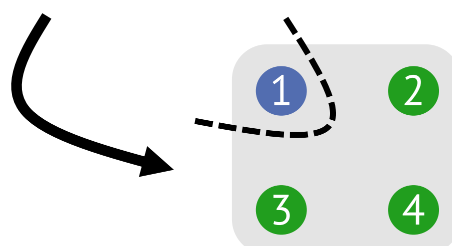
$\mathbf{w}_e(\{1\})$ = penalty for {1} vs. {2,3,4} split

Assumptions.

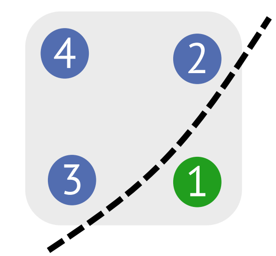
Uncut ignoring $\mathbf{w}_e(e) = \mathbf{w}_e(\emptyset) = 0$

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

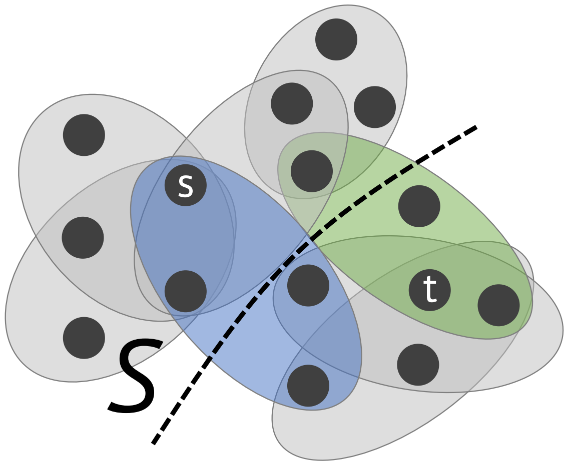
Symmetry $\mathbf{w}_e(U) = \mathbf{w}_e(e \backslash U)$ for all $U \subset e$.



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



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.

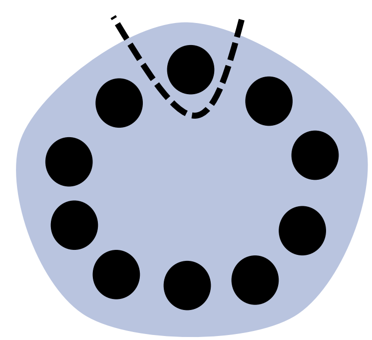
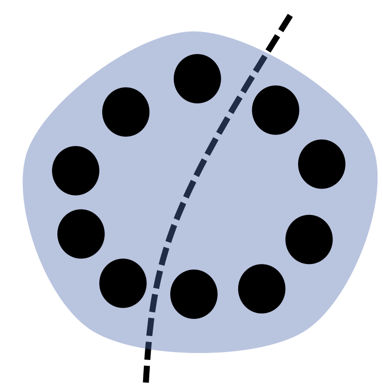
$$\text{minimize}_{S \subset V} \quad \sum_{e \in E} \mathbf{w}_e(e \cap S) \equiv \text{cut}_{\mathcal{H}}(S)$$
$$\text{subject to} \quad s \in S,\ t \notin S.$$

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.

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

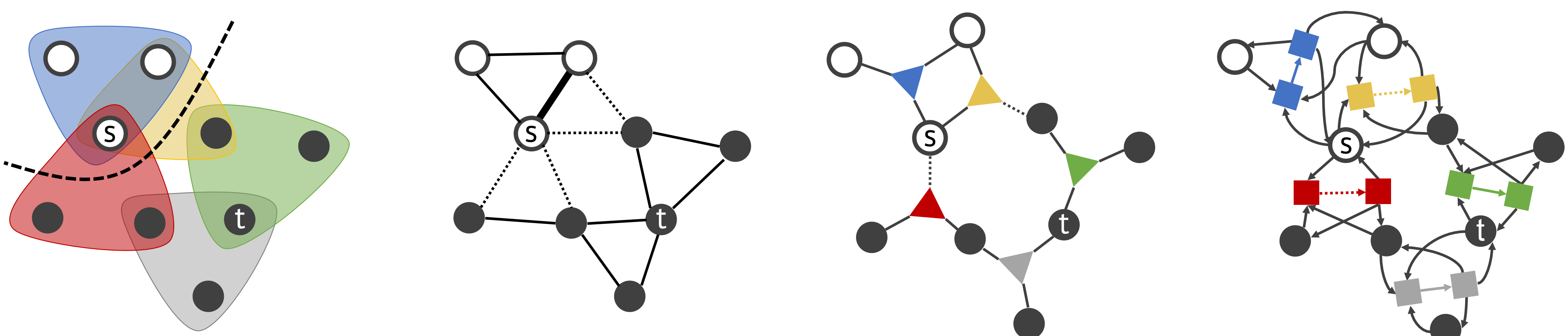
This allows us to distinguish between  and  .

We solve hypergraph cut problems with graph reductions.

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

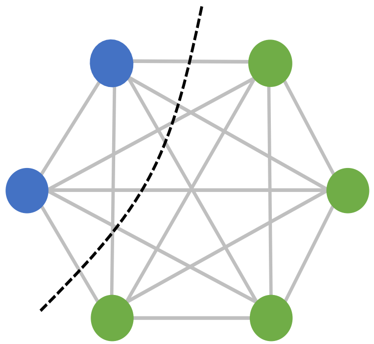
hyperedge | clique expansion | star expansion | Lawler gadget [1973]

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.

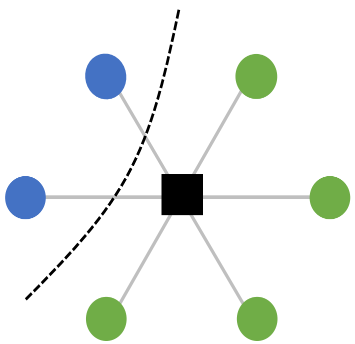


Clique Gadget
Does not require adding new vertices
[Agarwal+ 06; Zhou+ 06; Benson+ 16]

models →

Quadratic penalty

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

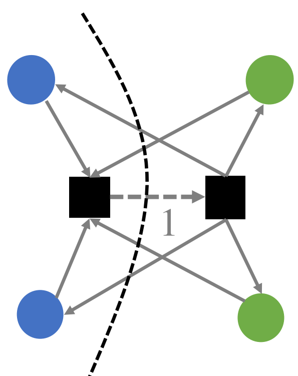


Star Gadget
Equivalent to bipartite expansion of a hypergraph
[Hu-Moerder 85; Heuer+ 18]

models →

Linear penalty

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



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.

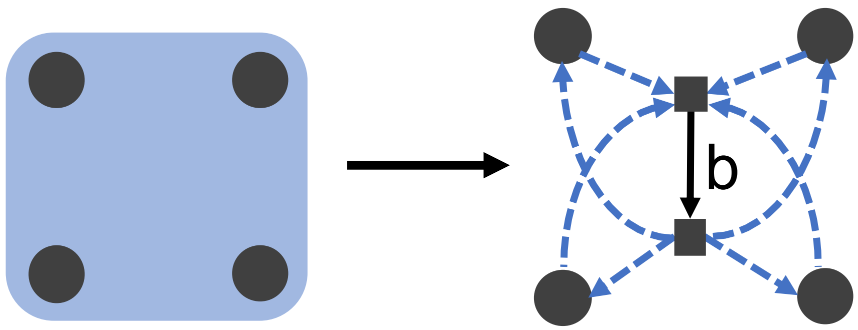
Discount cut $\mathbf{w}_e(U) = \min\{|U|^\alpha, |e \backslash U|^\alpha\}$ [Yaros-Imielinski 13]

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

This gadget models $\min(|U|, |e \setminus U|, b)$.

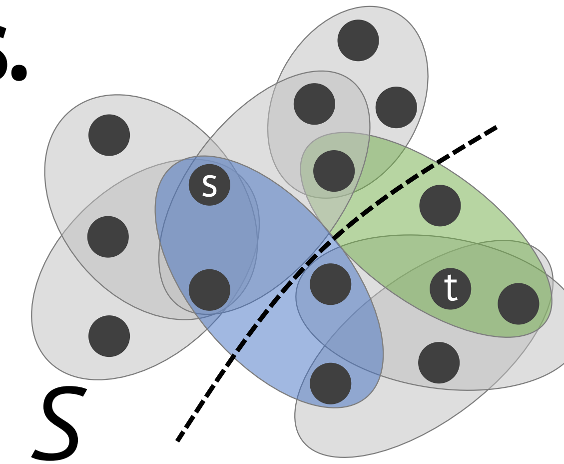
Theorem [Veldt-Benson-Kleinberg 20a]. Nonnegative linear combinations of the C-B gadget can model any submodular cardinality-based splitting function.

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

Essentially all 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.



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

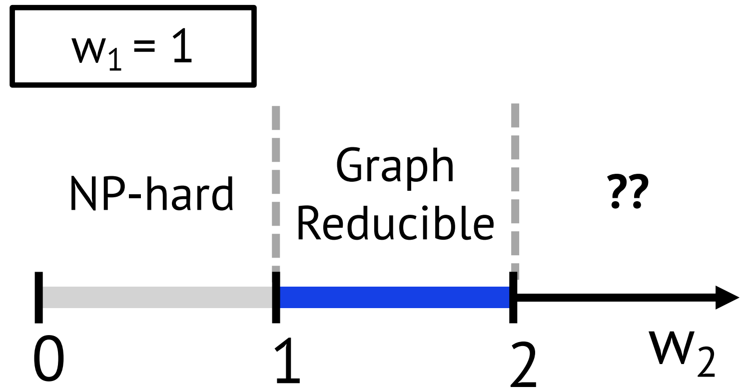
Cardinality-based splitting functions.

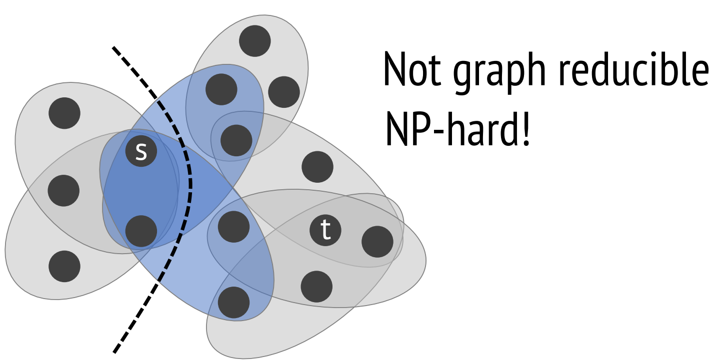
$\mathbf{w}_e(U) = f(\min(|U|, |e \backslash U|))$

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.

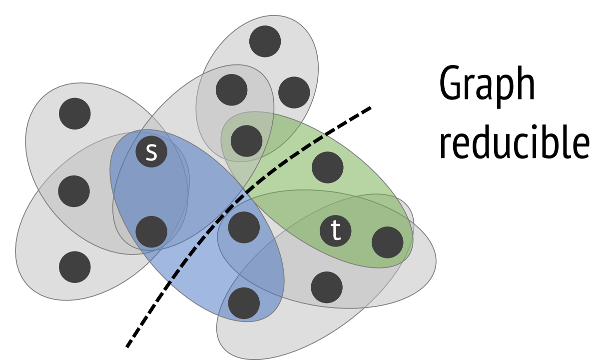
What happens when the splitting function isn't submodular?
Can we use some other algorithm?

Hardness and open questions for 4-node case.

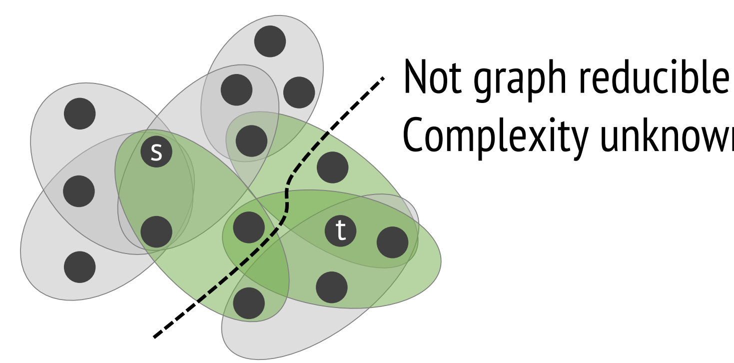




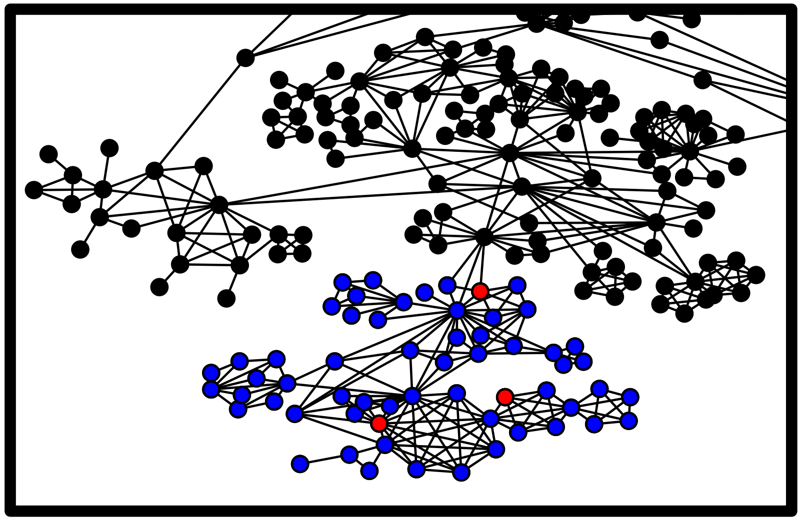
w_2 = 0.5 solution



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**

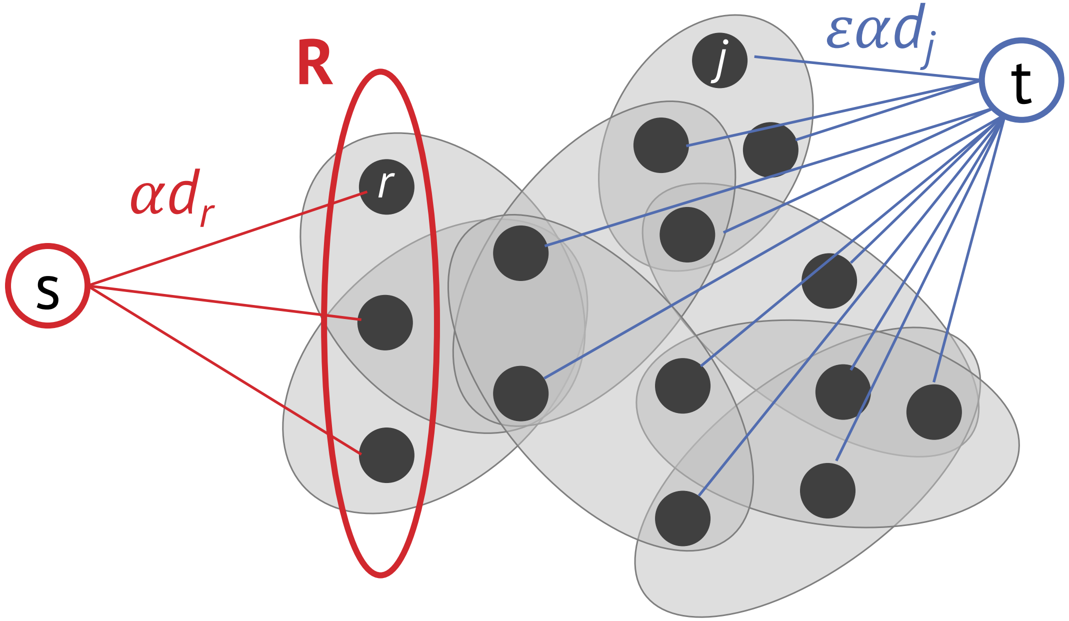
Examples.

Determine related products from co-purchasing data [Veldt-Benson-Kleinberg 20].

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

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

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



We introduce a new Hypergraph Local Conductance objective.

Hypergraph cut function

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

Encourage overlap with seed set.

Discourage overlap outside seed set

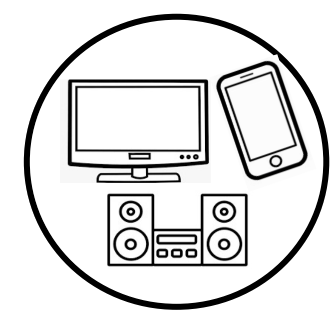
Theorem [Veldt-Benson-Kleinberg 2020b]

If $\text{cut}_{\mathcal{H}}(S)$ 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

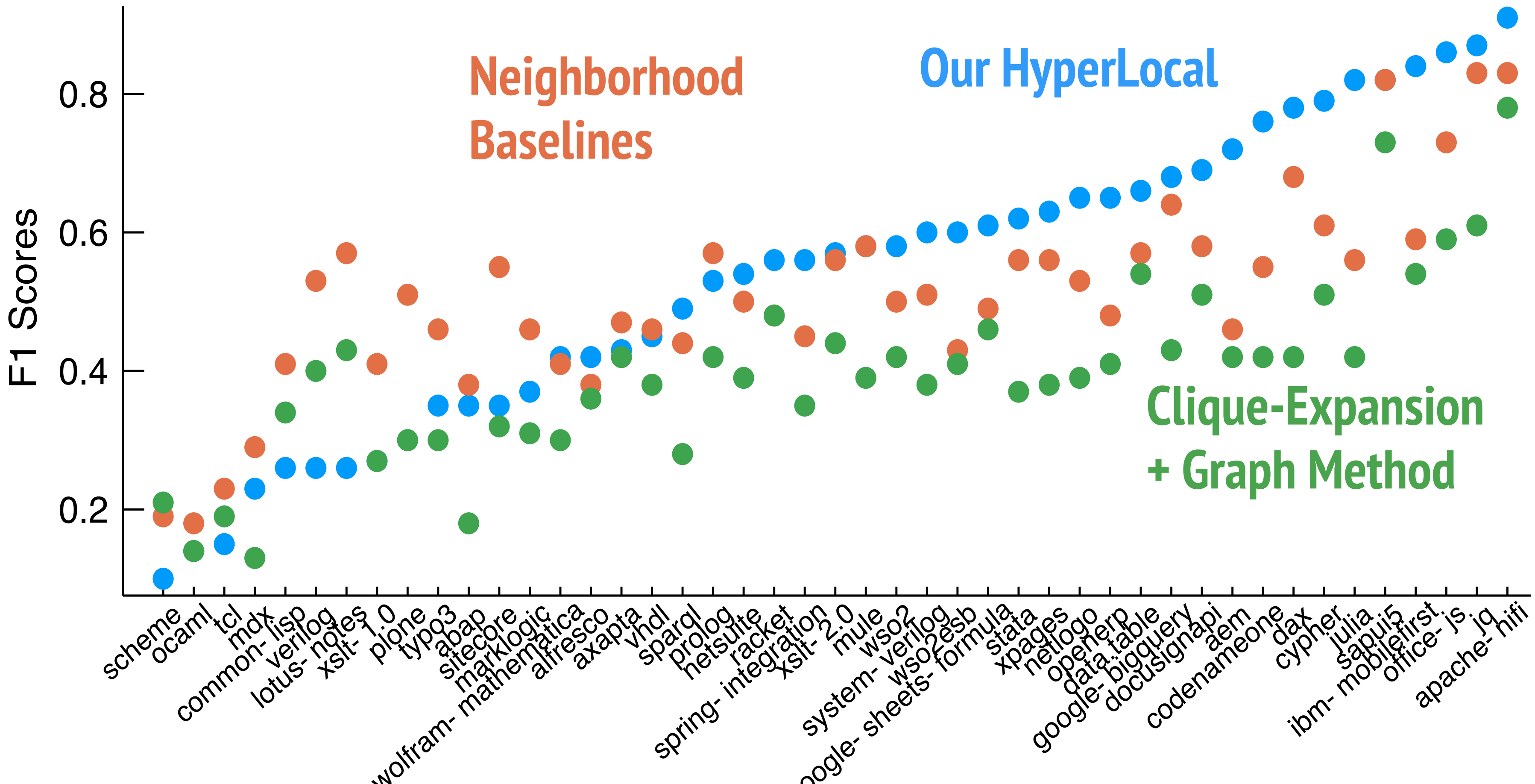
Cluster	$\|T\|$	time (s)	HyperLocal	Baseline1	Baseline2
Amazon Fashion	31	3.5	**0.83**	0.77	0.6
All Beauty	85	30.8	**0.69**	0.60	0.28
Appliances	48	9.8	**0.82**	0.73	0.56
Gift Cards	148	6.5	**0.86**	0.75	0.71
Magazine Subscriptions	157	14.5	**0.87**	0.72	0.56
Luxury Beauty	1581	261	**0.33**	0.31	0.17
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
- Product categories provide cluster labels
- All-or-nothing penalty ($w_i = 1$).

Max hyperedge size ~9.3k nodes!

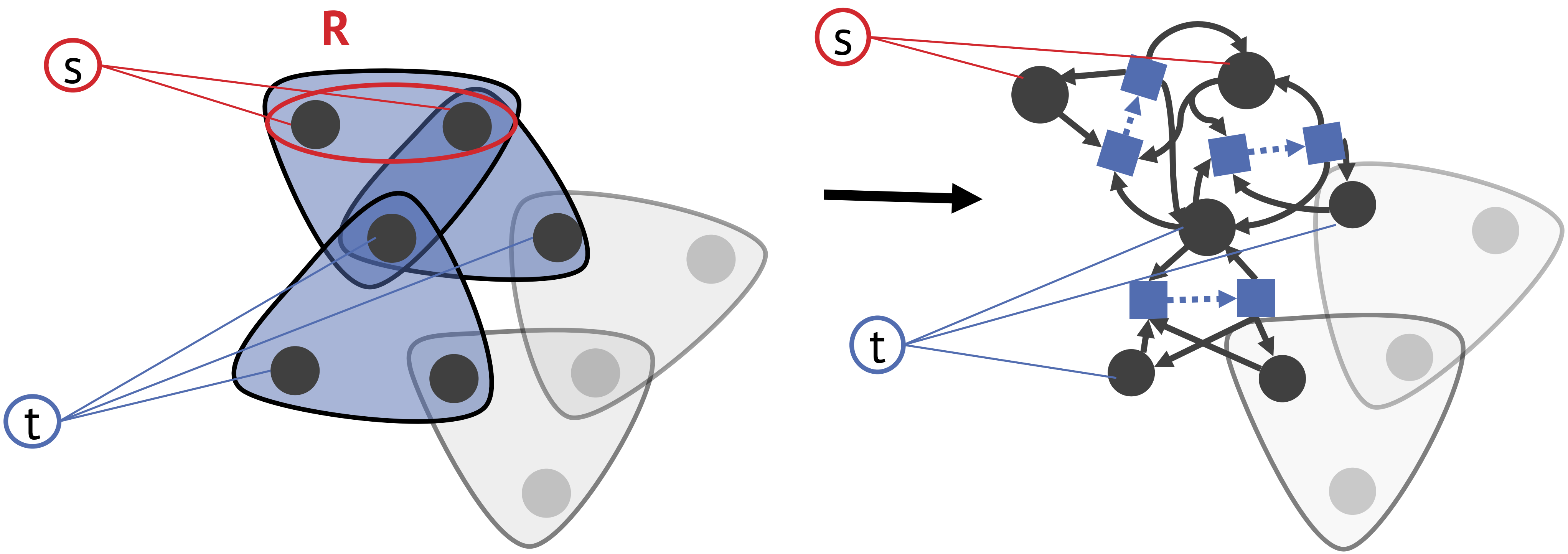
Detecting online forum questions on the same topic



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

Max hyperedge size ~60k nodes!

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



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.

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

THANKS! Austin Benson

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

@austinbenson

arb@cs.cornell.edu

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.
To appear at KDD, 2020.
github.com/nveldt/HypergraphFlowClustering

Augmented Sparsifiers for Generalized Hypergraph Cuts.
Nate Veldt, Austin R. Benson, and Jon Kleinberg.
arXiv:2007.08075, 2020.