Learning from networks with nonlinear eigenvectors: core-periphery detection in hypergraphs

Francesco Tudisco

Calcolo Scientifico e Modelli Matematici
Alla ricerca delle cose nascoste attraverso le cose manifeste

GSSI • Gran Sasso Science Institute • L'Aquila (Italy)

Thanks to

Desmond J. Higham
(Edinburgh, UK)

- T, Higham, *Node and edge eigenvector centrality for hypergraphs*, arXiv:2103.14867
- T, Higham, *Core-periphery detection in hypergraphs*, arXiv:2202.12769

Goal and outline

Goal

How to use “nonlinear eigenvectors” in core-periphery classification of hypergraphs

Outline

- Review of graph clustering via nonlinear eigenvectors
- Extension to hypergraphs
- Core-periphery classification of hypergraphs via nonlinear eigenvectors
- Numerical examples

“Nonlinear eigenvector”

Solution to $F(x) = \lambda x$, where:

- $F(x) =$ composition of linear and point-wise nonlinear mappings
  $= \sigma_1(A_1 \sigma_2(A_2 x))$
  Generalized linear models in deep learning

- $F(x) =$ action of a matrix valued operator
  $= M(x)x$
  Nonlinear eigenvalue problems with eigenvector nonlinearity

Nonlinear Laplacians

$B =$ incidence, boundary, gradient operator

$$L(x) = B\, g(B^\top f(x)) \qquad B : \text{edges} \to \text{nodes}$$

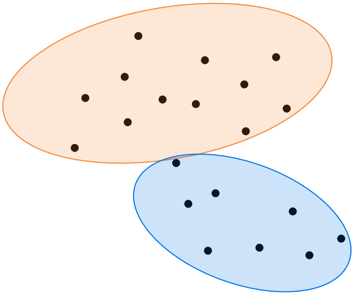
Different choices of f and g are used in several different settings:

- $f = \mathrm{Id}$, $g(x) = |x|^{p-1}\mathrm{sign}(x)$ . . . . graph p-Laplacian
  [Bühler&Hein, 2009], [Elmoataz et al, 2008], [Zhang, 2016], [T&Hein, 2018], ...
- exp and log . . . . consensus dynamics and chemical reactions
  [Neuhäuser et al, 2021], [Rao et al, 2013], [Van Der Schaft et al, 2016], ...
- Trigonometric functions . . . . network oscillators
  [Battiston et al, 2021], [Millán et al, 2020], [Schaub et al, 2016], ...
- Polynomials . . . . semi-supervised learning
  [Arya et al, 2021], [Ibrahim&Gleich, 2019], [Prokopchick et al, 2021], ...

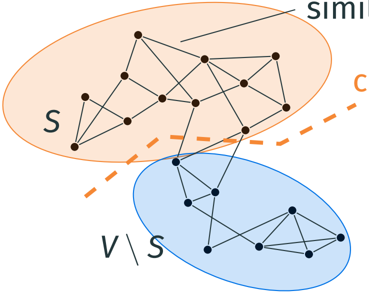
- p-norm-based . . . . centrality and core-periphery

(Hyper)graph classification

Unsupervised classification via graph clustering



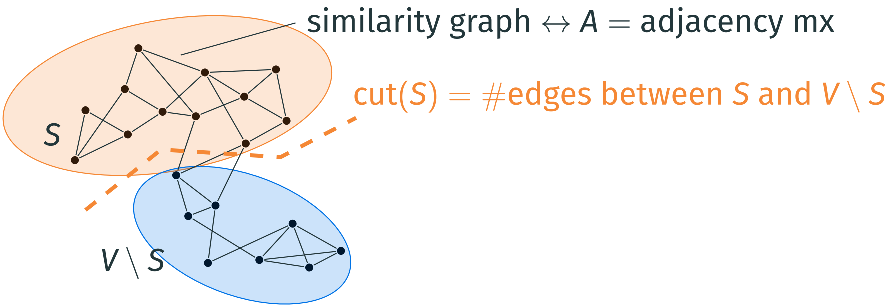
Unsupervised classification via graph clustering



similarity graph $\leftrightarrow A =$ adjacency mx

cut$(S) = \#$edges between S and $V \setminus S$

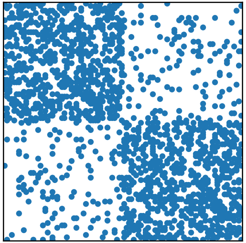
Unsupervised classification via graph clustering



Formulation as combinatorial optimization problem:

$$\min \left\{ K(S) = \mathrm{cut}(S)/|S| \;:\; S \subseteq \{1, \ldots, n\}, |S| \leq n/2 \right\}$$

Matrix reordering



Clusters C_1, C_2: many edges $C_i \leftrightarrow C_i$ and few edges $C_i \leftrightarrow C_j$ $(i \neq j)$

Extensions: discrete $\rightarrow$ continuous optimization

Extensions: discrete $\to$ continuous optimization

Fiedler extension: $\min \left\{ f(x) = \sum_{ij} A_{ij}(x_i - x_j)^2 \ : \ x^T \mathbb{1} = 0, \|x\|_2 = 1 \right\}$

- $\min f(x) \leq \min K(S) \leq \sqrt{2 \min f(x)}$
- It boils down to a matrix-eigenvalue problem which we know how to solve very efficiently to arbitrary precision

Extensions: discrete $\to$ continuous optimization

Fiedler extension: $\min \left\{ f(x) = \sum_{ij} A_{ij}(x_i - x_j)^2 \; : \; x^T \mathbb{1} = 0, \|x\|_2 = 1 \right\}$

- $\min f(x) \leq \min K(S) \leq \sqrt{2 \min f(x)}$
- It boils down to a matrix-eigenvalue problem which we know how to solve very efficiently to arbitrary precision

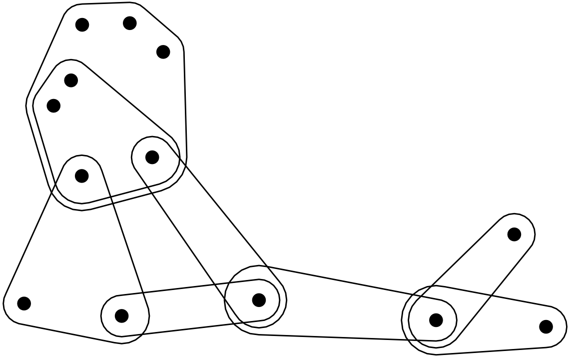
Lovász Extension: $\min \left\{ \ell(x) = \sum_{ij} A_{ij}|x_i - x_j| \; : \; x^T \mathbb{1} = 0, \|x\|_1 = 1 \right\}$

- $\min \ell(x) = \min K(S)$
- It can be interpreted as a nonlinear eigenvector problem, but it cannot be solved in polynomial time

How do we extend to hypergraphs?

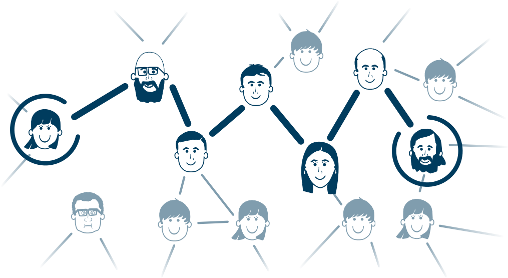
Hypergraph:

- $H = (V, E)$ where $e \in E$ can contain an arbitrary number of nodes
  H is a standard graph if $|e| = 2$, for all $e \in E$



Why hypergraphs?

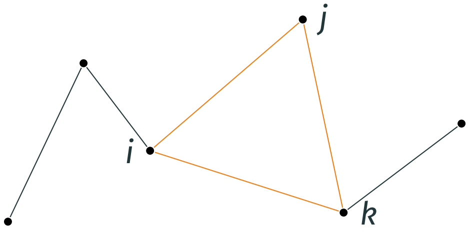
Relational data is full of interactions that happen in groups




Why hypergraphs?

Directly considering graph *motifs* brings a great deal of new insight

Example: Triangle hypergraph

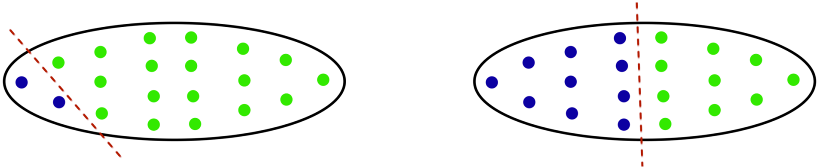


Hyperedge $e = \{ijk\} \in E$ if the graph contains all three edges ij, jk and ki

Hypergraph cut

How to quantify the *strength of a cut* in the hypergraph case?

Nice recent review: [Veldt,Benson,Kleinberg, SIREV, 2022]



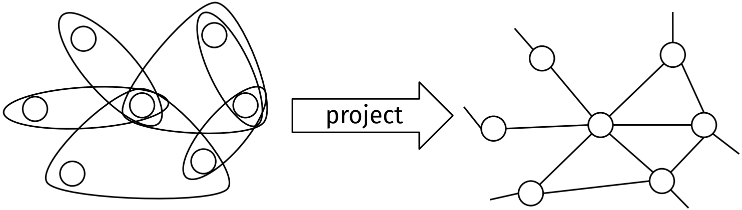
All-or-nothing hypergraph cut

Directly formulate a hypergraph cut by extending the notion of graph cut

- Graph:
  $\partial S = \{ij \in E : i \in S, j \in S^c\}$, $\text{cut}_G(S) = \sum_{ij \in \partial S} A_{ij}$
- All-or-nothing hypergraph cut:
  $\partial S = \{e \in E : e \cap S \neq \emptyset, e \cap S^c \neq \emptyset\}$, $\text{cut}_H(S) = \sum_{e \in \partial S} A_e$

Graph projection

A standard approach to represent and deal with hypergraphs



Example: Clique-expansion.

Form a graph where all nodes in an hyperedge are fully connected.

Cut on the clique-expanded graph: $\mathrm{cut}_{CE}(S) = \sum_{e \in \partial S} |e \cap S||e \cap S^c| A_e$

Hypergraph vs projected graph

How do cut_{CE} and cut_{H} compare?

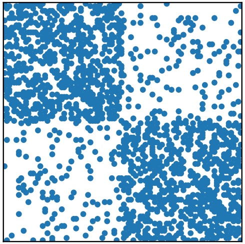
Semi-supervised classification with $\sim$ 2% input labels

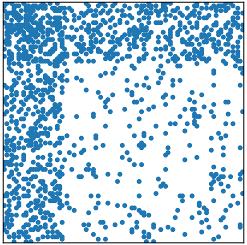
		Rice31 $n = 3560$ $c = 9$	Caltech36 $n = 590$ $c = 8$	FMNIST $n = 60000$ $c = 10$
cut_{H}	Lovász	82.6 ± 0.1	66.1 ± 0.3	79.7 ± 1.1
cut_{CE}	Fiedler	80.5 ± 2.7	55.3 ± 2.9	70.5 ± 3.3

Criticism: the additional complication of H is not justified

Core-periphery classification

Clustering vs core-periphery





Clusters C_1, C_2: many edges $C_i \leftrightarrow C_i$ and few edges $C_i \leftrightarrow C_j$ $(i \neq j)$

Core-periphery C, P: many edges $C \leftrightarrow C$ and $C \leftrightarrow P$, few $P \leftrightarrow P$

Core–periphery in networks

Borgatti, Everett, *Social Networks*, 1999

Core/periphery structure in a citation network

	16S	7C	13J	15S	17S	4C	1A	8C	9F	10I	11J	2A	3B	14P	5C	6C	12J	18S	19S	20S
16 SSR	1	1	1	1	1			1				1				1	1	1		1
7 CYSR	1	1		1	1	1			1					1	1	1				1
13 JSWE	1		1	1	1							1	1			1		1	1	1
15 SCW	1	1	1	1	1	1		1	1		1	1			1	1		1	1	1
17 SW	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1		1	1	1
4 CAN		1		1	1	1								1		1				
1 AMH					1		1													
8 CSWJ	1			1	1			1												
9 FR		1		1	1				1											
10 IJSW					1					1										
11 JGSW				1	1						1									
2 ASW	1		1	1	1							1		1						1
3 BJSW			1		1								1					1		
14 PW		1			1	1						1		1		1				
5 CCQ		1		1											1	1				
6 CW	1	1	1	1	1	1								1	1	1				1
12 JSP	1																1			
18 SWG	1		1	1	1								1					1	1	
19 SWHC			1	1	1													1	1	
20 SWRA	1	1	1	1	1							1				1				1

Core: nodes strongly connected across the whole network

Periphery: nodes strongly connected only to the core

Combinatorial optimization formulation

Writing the problem in terms of cut(S) leads to a two-variable combinatorial problem, for which the Lovász approach does not work

Matrix reordering formulation:

Find the permutation $i \mapsto p_i$ that solves

$$\max \left\{ \sum_{ij} A_{ij} \max\{p_i, p_j\} \; : \; p = \text{permutation of } \{1, \ldots, n\} \right\}$$

Coreness score relaxation

Relax the constraint $p =$ permutation into *"nonnegative with fixed norm"*

$$\max \left\{ \sum_{ij} A_{ij} \max\{x_i, x_j\} \ : \ x \geq 0, \|x\| = 1 \right\}$$

Coreness score relaxation

Relax the constraint $p =$ permutation into *"nonnegative with fixed norm"*

$$\max\left\{\sum_{ij} A_{ij} \max\{x_i, x_j\} \;:\; x \geq 0, \|x\| = 1\right\}$$

Use a "softmax": for $e = (i, j)$ and $x|_e = (x_i, x_j)$ we have $\max\{x_i, x_j\} = \|x|_e\|_\infty$

$$\max\left\{\sum_e A_e \|x|_e\|_p \;:\; x \geq 0, \|x\| = 1\right\} \qquad (p \text{ large})$$

We obtain a model for coreness score: "$x_i > x_j$ if i is more in the core than j"

Hypergraph case

All-or-nothing core-periphery model:

e is a "good" hyperedge if it contains at least one core node

$x|e = (x_{i_1}, \ldots, x_{i_{|e|}})$

$$\max \left\{ \varphi(x) = \sum_{e \in E} A_e \, \| x|_e \|_p \quad \text{s.t.} \quad x \geq 0, \|x\| = 1 \right\}$$

Questions for the remaining slides

- How can we solve $\max \varphi(x)$?
- How does it compare with doing core-periphery classification on the projected graph?

Nonlinear eigenvector formulation

Let $\|\cdot\| = \|\cdot\|_q$ be the norm defining the constraint.

$\varphi(\alpha x) = \alpha\varphi(x) \Rightarrow$ look at the unconstrained problem $\max_x \varphi(x)/\|x\|_q$

$$\nabla\left\{\frac{\varphi(x)}{\|x\|_q}\right\} = 0 \iff B\,g(B^\top f(x)) = \lambda x$$

- B : hyperedges $\to$ nodes hypergraph incidence operator
- $f(x) = x^{\frac{p}{p-q}}$ (entrywise)
- $g(x) = x^{\frac{1}{q-1}}$ (entrywise)

Nonlinear eigenvalue formulation (cont.)

$$\max \left\{ \varphi(x) = \sum_{e \in E} w(e) \, \| x|_e \|_p \quad \text{s.t.} \quad x \geq 0, \|x\|_q = 1 \right\}$$

is equivalent to

$$L(x) = B\, g(B^\top f(x)) = \lambda x, \qquad x \geq 0$$

Flavor of Perron-Frobenius problem:
we look for a nonnegative solution which is maximal (in some sense)

Main result

If $p > q > 1$, there exists a unique nonnegative eigenvector x^* of $L(x)$.

Moreover:

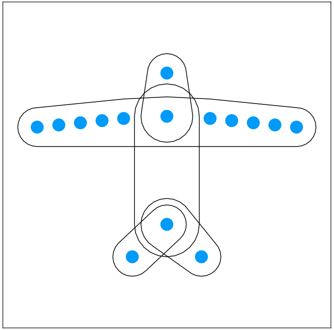
- x^* is entrywise positive
- the iterative method
  - $y \leftarrow \mathrm{Diag}(x)^{q-1} B \left(B^T x^q\right)^{\frac{1}{q}-1}$
  - $x \leftarrow \left(y/\|y\|_{p^*}\right)^{\frac{1}{p-1}}$, $\quad p^* =$ dual norm of p

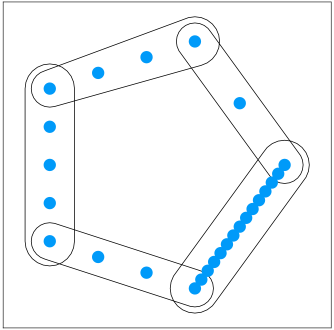
  converges to x^* for any positive starting point $x^{(0)}$, with linear rate of convergence $O(|p-1|/|q-1|)$.

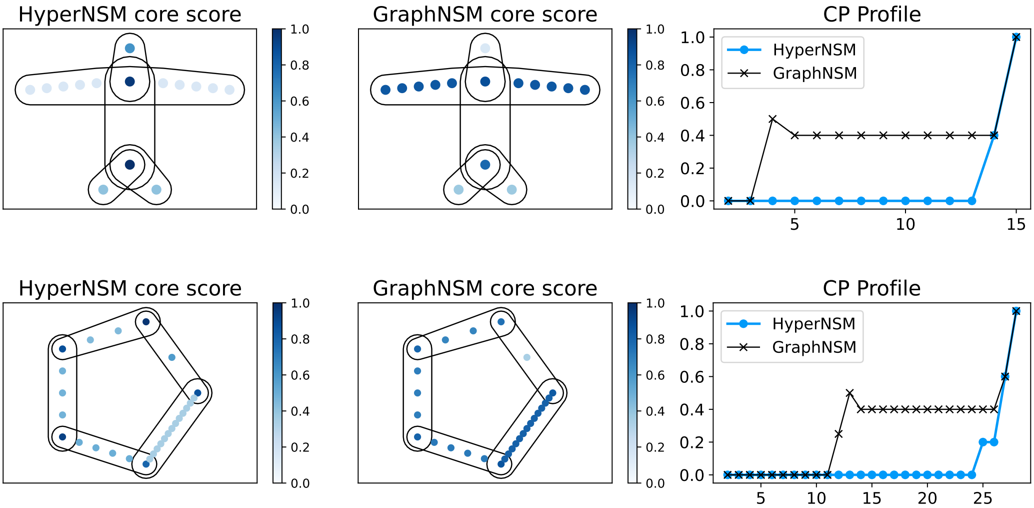
Note: cost per iteration $= O(B, B^T \times \text{vector})$

G vs H: Examples

Examples: hyperplane and hypercycle







HyperNSM core score
GraphNSM core score
CP Profile
HyperNSM
GraphNSM
1.0
0.8
0.6
0.4
0.2
0.0
5
10
15
20
25

Core-periphery profile

Extension of the core-periphery profile for graphs [DellaRosa et al, 2013]

For any subset of nodes $S \subseteq V$ consider the quantity

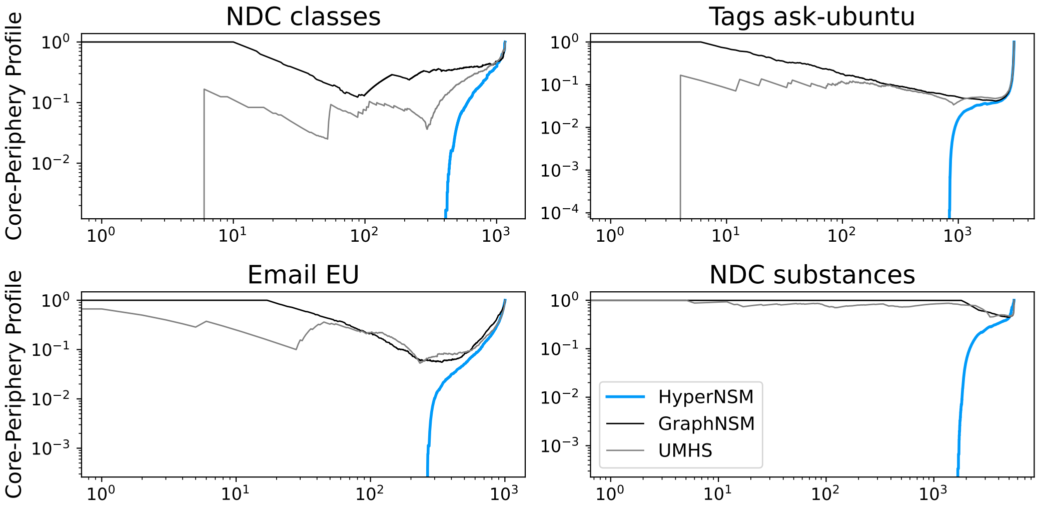
$$\gamma(S) = \frac{\text{\# edges all contained in } S}{\text{\# edges with at least one node in } S}$$

Hypergraph core-periphery profile:

function $\gamma_x(k)$ that to any $k \in \{1, \dots, n\}$ associates the value $\gamma(S_k(x))$ where $S_k(x)$ is the set of k nodes with smallest coreness score in x

$\gamma(S)$ is small if S is largely contained in the periphery of the hypergraph

Real-world datasets



Conclusions and questions

- When dealing with classification problems on graphs and hypergraphs we end up nonlinear eigenvector problems
- One example is core-periphery. Here, unlike clustering:

Conclusions and questions

- When dealing with classification problems on graphs and hypergraphs we end up nonlinear eigenvector problems
- One example is core-periphery. Here, unlike clustering:
  - we have guarantees for uniquenes and computation of the solution to the nonlinear eigenvector problem
  - the results are very different on the projected graph

Conclusions and questions

- When dealing with classification problems on graphs and hypergraphs we end up nonlinear eigenvector problems
- One example is core-periphery. Here, unlike clustering:
  - we have guarantees for uniquenes and computation of the solution to the nonlinear eigenvector problem
  - the results are very different on the projected graph

¿Q? What are other settings where graph projections fail?

¿Q? What is a tight convergence rate for the iterative method?

¿Q? Is there a better method than (nonlinear) power method?

Conclusions and questions

- When dealing with classification problems on graphs and hypergraphs we end up nonlinear eigenvector problems
- One example is core-periphery. Here, unlike clustering:
  - we have guarantees for uniquenes and computation of the solution to the nonlinear eigenvector problem
  - the results are very different on the projected graph

¿Q? What are other settings where graph projections fail?
¿Q? What is a tight convergence rate for the iterative method?
¿Q? Is there a better method than (nonlinear) power method?

Thank you!