Statistical modeling of higher-order interactions - towards node clustering

Catherine Matias

Centre National de la Recherche Scientifique, Paris, France
Sorbonne Université, Paris, France
Université de Paris Cité, Paris, France

Autumn school on hypergraphs - Oct 2023

Outline

Higher-order interactions I

Motivations

- Networks or graphs focus on pairwise interactions
- These type of pairwise interactions can already be quite elaborate: undirected/directed, binary/weighted, simple/multiple, static/dynamic, multiplex or multi-layers, ...
- Nonetheless pairwise interactions are not sufficient to describe the nature of complex interactions:
  - e.g. the presence of a 3rd chemical component may modify the type of interaction of 2 other;
  - e.g. a collaboration between 3 authors is stg different from 3 pairwise collaborations between these same authors;
- Collective interactions or group interactions are richer than just pairwise interactions

$\hookrightarrow$ These are called higher-order interactions (HOI).

Higher-order interactions II

Where do we find HOI?

- Social networks: triadic and larger groups (as early as Simmel, 1950)
- Scientific co-authorship,
- Interactions between chemical components,
- or species in ecological systems,
- neurons in brain networks,
- etc

These interactions **CAN NOT** be represented by a graph.

Higher-order interactions III

This is a nice recent review (2020):

Contents lists available at ScienceDirect

Physics Reports

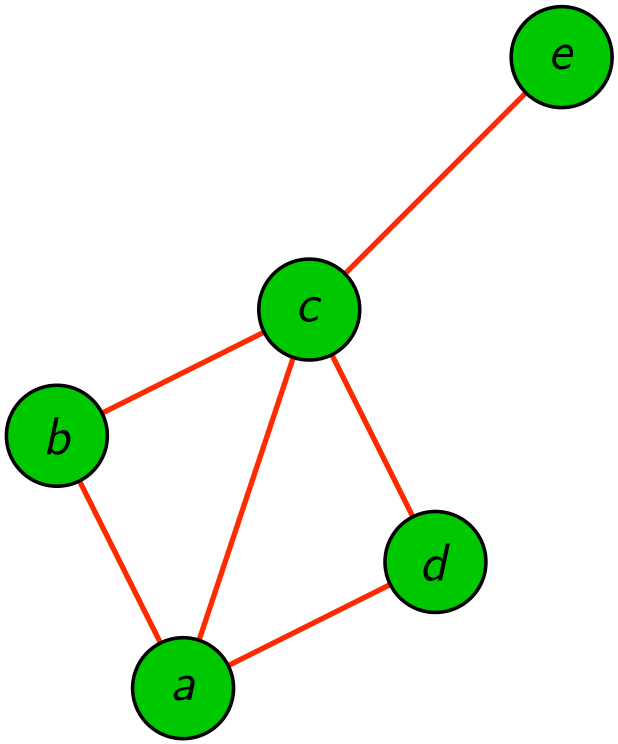
journal homepage: www.elsevier.com/locate/physrep

Networks beyond pairwise interactions: Structure and dynamics

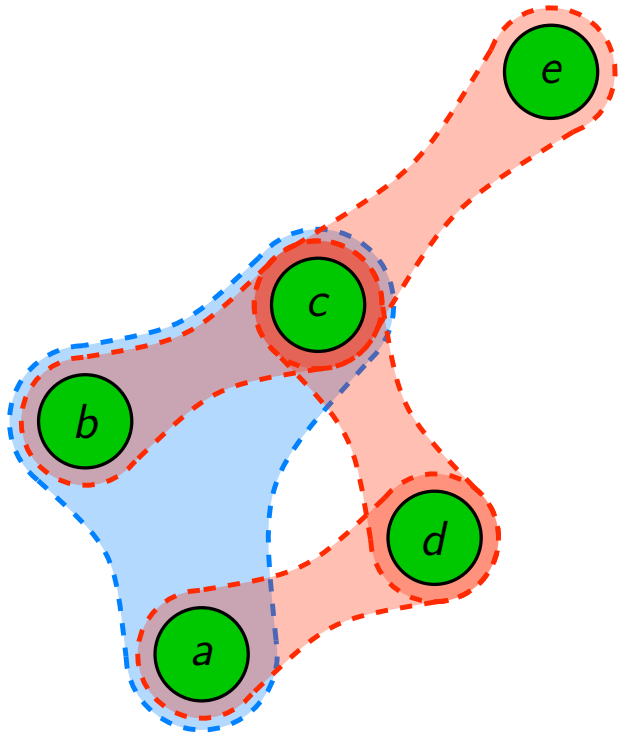
Federico Battiston [a,*], Giulia Cencetti [b], Iacopo Iacopini [c,d], Vito Latora [c,e,f,g], Maxime Lucas [h,i,j], Alice Patania [k], Jean-Gabriel Young [l], Giovanni Petri [m,n]

Pairwise vs HOI

HOI are defined as sets of interacting entities.
e.g. $V = \{a, b, c, d, e\}; \mathcal{I} = \{\{a, b, c\}, \{a, d\}, \{c, d\}, \{c, e\}\}$



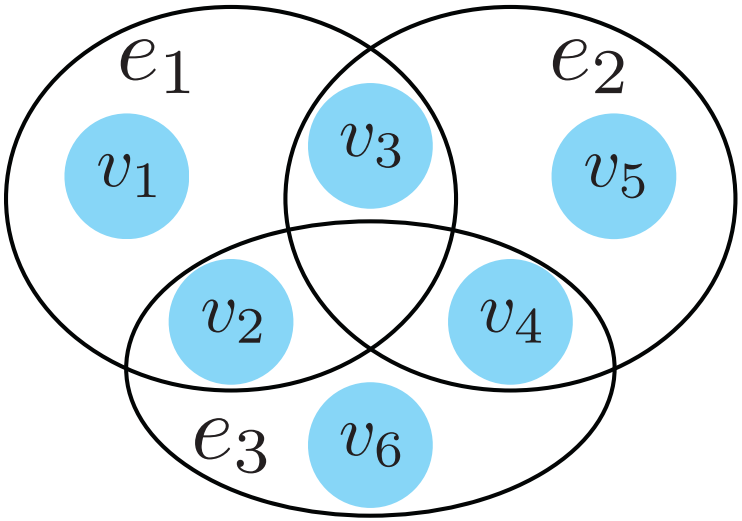
(a) Pairwise interactions

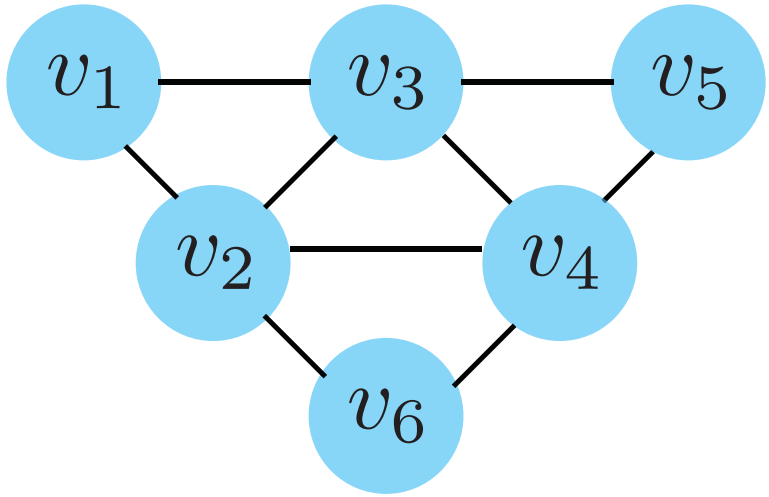


(b) A HOI in blue

Outline

Naïve Graph representation: clique reduction graph

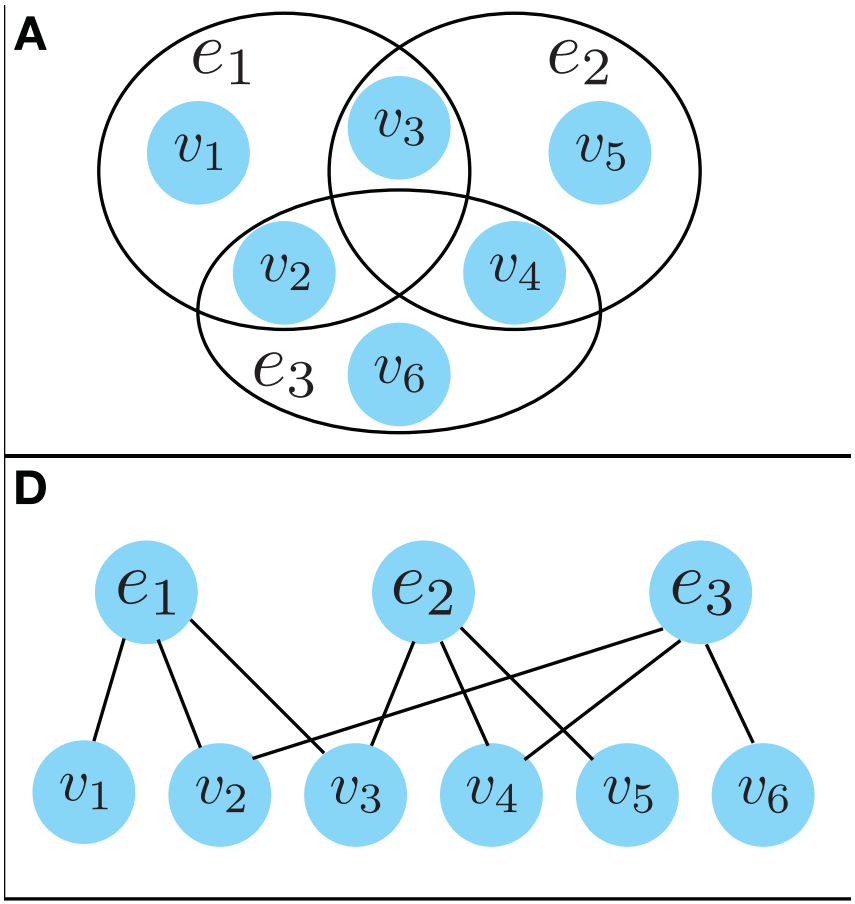




Picture from Schaub *et al.* 2021

- Each interaction is transformed into a clique = all edges between pairs are present ;
- HOIs actually disappeared !
- Too simplistic: For e.g, in co-authorship 1 paper with 3 authors $\neq$ 3 different papers written by pairs of those authors.

Bipartite graph representation (two-modes network or star-expansion graph)



- No loss of information;
- But "higher-order" now translates into node degrees in one part;
- 2 two parts don't play symmetric roles: statistical models on bipartite graphs are not appropriate here

Picture from Schaub *et al.* 2021

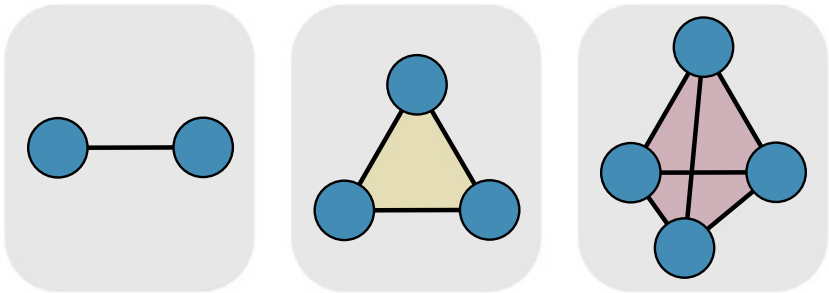
Other graph representations

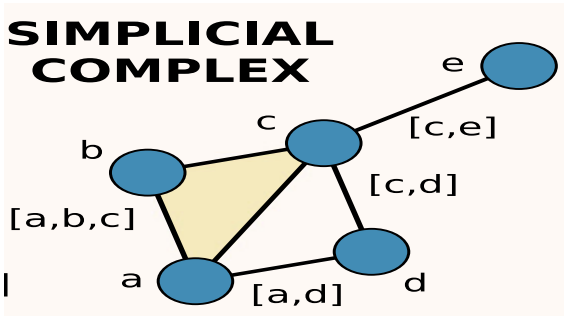
- There are other graph-representations of HOIs
- But none of it may completely capture these

$\hookrightarrow$ There are 2 mathematical objects to represent HOIs : Simplicial complexes and hypergraphs.

Simplicial complexes vs hypergraphs I

Picture from Battiston *et al.*





Simplex and Simplicial complexes

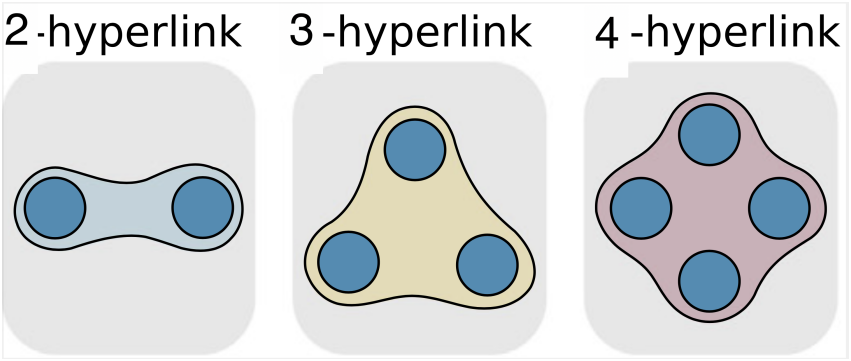
- a k-simplex $\sigma = \{p_0, p_1, \ldots, p_k\}$ is a set of $k+1$ points (in a topological space);
- a subface of a simplex σ is any subset of points in σ;
- a simplicial complex = a collection $K = \{\sigma_1, \ldots, \sigma_n\}$ of simplexes (of any size);
- a valid simplicial complex is such that $\forall \sigma \in K$, every subface of σ also belongs to K

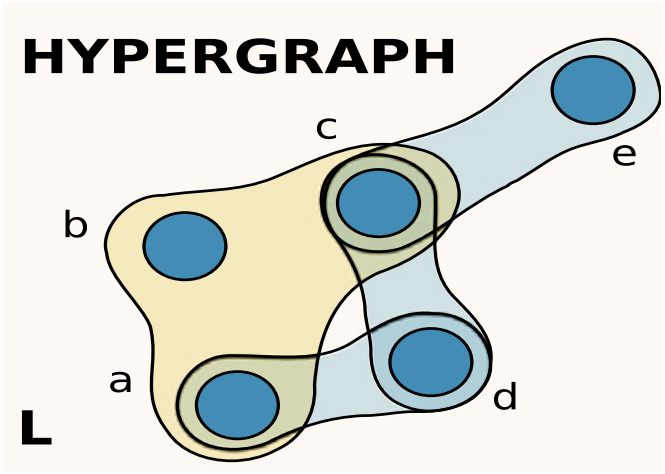
Simplicial complexes vs hypergraphs II

(Dis)-Advantages

- ☺ strong mathematical object, very useful in many areas; e.g: statistical topological data analysis, to approximate varieties of irregular algebraic structures;
- ☹ Valid simplicial complexes impose all sub-interactions of an interaction should exist;
- ☹ points come with positions in (topological) space

Simplicial complexes vs hypergraphs III





Definition

A hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ is defined as a set of nodes $\mathcal{V} \neq \emptyset$ and a set of hyperedges $\mathcal{E}$. **Each hyperedge is a non-empty collection of k distinct nodes** taking part in an interaction.

Simplicial complexes vs hypergraphs IV

Hypergraphs characteristics

- Hypergraphs naturally include the entity of graphs, by simply considering hyperedges of size $k = 2$;
- A hypergraph may contain a size-3 hyperedge $\{a, b, c\}$ without any requirement on the existence of the size-2 hyperedges $\{a, b\}$, $\{a, c\}$, and $\{b, c\}$.

Simplicial complexes vs hypergraphs V

Simple hypergraphs and variants

- In **simple hypergraphs**, an hyperedge appears only once and contains distinct nodes;
- May consider **nodes to appear with multiplicities** in a same hyperedge
  - Example: chemical reactions, multiplicity = stoichiometric coefficient;
  - I call these **multisets** hypergraphs;
  - generalize (in some sense) the notion of loops in graphs
- May consider **multiple** hyperedges, when a same hyperedge may appear several times (= integer-valued weight on a hyperedge);
- May introduce a **direction**: a hyperedge e is divided into 2 ordered subsets (e_1, e_2) of interacting nodes ($e = e_1 \cup e_2$);
  $\hookrightarrow$ not much used though;

NB : in the following, focus on hypergraphs.

Matrix encoding of HOIs

- **Incidence matrix** H, size $n \times m$ where n nb of nodes, m nb of interactions; with entry $H_{i,e} = 1$ when node i belongs to hyperedge e.
  $\hookrightarrow$ contains all the information;
  $\hookrightarrow$ enables definition of **node degrees** d_i (=rowSums of H) and **hyperedge sizes** δ_e (=colSums of H)
- **Reduced adjacency matrix:** $A = HH^\intercal - D$ has size $n \times n$, where $D = diag(d_1, \ldots, d_n)$
  $\hookrightarrow$ This is the adjacency matrix of the clique reduction graph;
  $\hookrightarrow$ contains only partial information;
- **Reduced adjacency with hyperedge size information:** Let $\Delta = diag(|e|)_{e \in \mathcal{E}}$. Some authors have considered different re-weightings of the clique reduction graph: $A' = H\Delta^{-1}H^\intercal$ or $A'' = H(\Delta - I)^{-1}H^\intercal$ (more on that later).

Outline

Statistical measures on HOIs

Graph statistics generalized to HOIs

- For any size $k \geq 2$, size-k density is = nb of size-k hyperedges $/\binom{n}{k}$
- Node degree; hyperedge size;
- **Centrality measures**
  - relies on the notion of paths;
  - a path is a sequence $(e_1, e_2, \ldots, e_t)$ of hyperedges such that 2 successive hyperedges have at least one common node ($e_i \cap e_{i+1} \neq \emptyset$);
  - concept of k-path: any 2 successive hyperedges share at least $k \geq 1$ nodes;

Graph statistics with no natural generalization

- clustering and transitivity (based on triangles);
- motifs (combinatorial complexity)

Outline

What are we looking for?

- In graphs, the concept of **communities** (aka nodes with high within-group and low between-groups connections) has been a driving concept for node clustering;
- In hypergraphs, when 2 nodes are 'connected', they might share this connection with other nodes (ie they belong to the same hyperedge of size $k \geq 2$). Should the sizes of the connections play a role? Should we focus on the percentage of nodes within the same group in the same hyperedge? Could we measure something else?
- From a statistical point of view, **clusters are more general than communities**: they describe sets of nodes that behave similarly in their connections; e.g. clusters of hubs, of peripheral nodes, ...
- What clusters that are not communities might look like in hypergraphs?

These are somehow still open questions.

Outline

Clustering the nodes of a hypergraph I

Some techniques: graph-based, spectral, modularity, Stochastic Blockmodels (SBM)

Graph reduction + hyperedge size information

- $A'' = H(\Delta - I)H^\intercal$ is a weighted graph reduction that preserves the nodes degrees. Kumar *et al.* (2020) propose an algorithm to maximize **Newman-Girvan modularity** on this graph A'' (Iteratively Reweighted Modularity Maximization (IRMM) alternates Louvain algorithm on a weighted clique reduction graph and new hyperedge weights computation). Shortcomings:
  - ☹ the exact composition of each hyperedge in nodes falling into the different clusters is captured only through pairs of nodes.

Clustering the nodes of a hypergraph II

Graph reduction + hyperedge size information (cont.)

- **Hypergraph Laplacian** $L = I - D^{-1/2} H \Delta^{-1} H^\intercal D^{-1/2}$ corresponds to Laplacian of $A' = H\Delta^{-1}H^\intercal$. Ghoshdastidar & Dukkipati (2014,2017) propose a spectral approach (Compute leading eigenvectors and run k-means on rows).
  Shortcomings:
  - ☹ works only if clusters may be identified from the weighted reduction graph
  - ☹ tend to favour groups of the same size
  - ☹ no criterion to select the number of groups

Clustering the nodes of a hypergraph III

Modularity-based approaches

- See for e.g. Chodrow *et al.*, 2021 and Kamiński *et al.*, 2019.
- Different hypergraph modularity defs: what kind of communities do they favour?
- For computational reasons, these focus on ***multisets-hypergraphs*** where nodes may be repeated in a same hyperedge;
- This is not always appropriate, e.g. co-authorship dataset;
- In the context of graphs, it is known that this is not the correct way to do it in the absence of self-loops and multiple edges. Question: what are the implications for hypergraphs?
- Other Shortcomings:
  - ☹ look for *communities* and not general clusters (e.g. hubs, peripheral nodes, disassortative behaviours, ...);
  - ☹ Modularity maximization is difficult; only local maximum is found;
  - ☹ No statistical criterion to select the number of groups.

Clustering the nodes of a hypergraph IV

Our SBM proposal (joint work with Luca Brusa)

- We focus on **simple** graphs (instead of multisets-hypergraphs);
- We define **a stochastic blockmodel** to cluster the nodes of a hypergraph
  - We establish **parameter identifiability** results;
  - We propose a **variational expectation-maximisation** algorithm to infer clusters and parameters;
  - We propose an **ICL criterion** to select the number of clusters;
  - All these tools are implemented (in C++) in a efficient `R` **package** called `HyperSBM` (`https://github.com/LB1304/HyperSBM`).

Outline

SBM formulation

- $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{1, \ldots, n\}$ nodes and $\mathcal{E}$ hyperedges;
- For each $2 \leq m \leq M$, let $\mathcal{V}^{(m)} = \left\{\{i_1, \ldots, i_m\} : i_1, \ldots, i_m \in \mathcal{V} \text{ and } i_1 \neq \ldots \neq i_m\right\}$, set of unordered node tuples of size m;
- **Observations:** At each $\{i_1, \ldots, i_m\} \in \mathcal{V}^{(m)}$, we observe indicator variable $Y_{i_1,\ldots,i_m} = 1\{\{i_1, \ldots, i_m\} \in \mathcal{E}\}$;
- **Latent clusters:** $Z_1, \ldots, Z_n$ iid in $\{1, \ldots, Q\}$ with $\pi_q = \mathbb{P}(Z_i = q)$;
- **Conditional independence assumption**: $\{Y_{i_1,\ldots,i_m}\}_{\{i_1,\ldots,i_m\} \in \mathcal{V}^{(m)}} | \{Z_1, \ldots, Z_n\}$ are independent with $Y_{i_1,\ldots,i_m} | \{Z_1 = q_1, \ldots, Z_m = q_m\} \sim \text{Bern}(B^{(m)}_{q_{i_1},\ldots,q_{i_m}})$.

Parameter (generic) identifiability

Generic identifiability: a parameter θ almost surely (*w.r.t.* Lebesgue measure) uniquely defines the distribution $\mathbb{P}_\theta$ (up to label switching on the node groups).

Theorem

For any Q, the parameter $\theta = (\pi_q, B^{(m)}_{q_1,\dots,q_m})_{m,q,q_1,\dots,q_m}$ of the HSBM for (simple) hypergraphs over n nodes, is generically identifiable for large enough n.

Said differently, there is a finite set $\mathcal{C}$ of (non explicit) polynomial conditions on θ such that whenever $\theta \notin \mathcal{C}$, the distribution $\mathbb{P}_\theta$ is uniquely defined by θ.

Inference through variational EM I

- Direct computation of the likelihood is not feasible for large n;
- EM algorithm neither feasible because latent variables are not independent conditional on observed ones;
- Variational approximation to EM algorithm: replace the intractable posterior distribution by the best approximation (w.r.t. Kullback-Leibler divergence) in a class of simpler (factorised) distributions:

$$\mathbb{Q}_\tau(Z_1, \ldots, Z_n) = \prod_{i=1}^{n} \mathbb{Q}_\tau(Z_i) = \prod_{i=1}^{n} \prod_{q=1}^{Q} \tau_{iq}^{Z_{iq}},$$

with the variational parameter $\tau_{iq} = \mathbb{Q}_\tau(Z_i = q) \in [0, 1]$ and $\sum_{q=1}^{Q} \tau_{iq} = 1$, for any $i = 1, \ldots, n$ and $q = 1, \ldots, Q$.

Inference through variational EM II

Evidence lower bound (ELBO)

$$\begin{aligned}\mathcal{J}(\theta,\tau) &= \mathbb{E}_{\mathbb{Q}_\tau}[\log \mathbb{P}_\theta(\boldsymbol{Y},\boldsymbol{Z})] - \mathbb{E}_{\mathbb{Q}_\tau}[\log \mathbb{Q}_\tau(\boldsymbol{Z})] \\ &= \log \mathbb{P}_\theta(\boldsymbol{Y}) - \mathsf{KL}(\mathbb{Q}_\tau(\boldsymbol{Z})||\mathbb{P}_\theta(\boldsymbol{Z}|\boldsymbol{Y})) \\ &\leq \log \mathbb{P}_\theta(\boldsymbol{Y}),\end{aligned}$$

with equality iff $\mathbb{Q}_\tau(\boldsymbol{Z})$ is the true posterior $\mathbb{P}_\theta(\boldsymbol{Z}|\boldsymbol{Y})$.

VEM maximises the lower bound $\mathcal{J}(\theta,\tau)$ (with respect to τ and θ) instead of the intractable log-likelihood $\log \mathbb{P}_\theta(\boldsymbol{Y})$

VEM algorithm

- **VE-Step** maximizes $\mathcal{J}(\theta, \tau)$ with respect to τ:

$$\widehat{\tau}^{(t)} = \arg\max_{\tau} \mathcal{J}(\theta^{(t-1)}, \tau); \quad \text{s.t. } \sum_{q=1}^{Q} \tau_{iq} = 1 \qquad \forall i = 1, \ldots, n.$$

  This is equivalent to minimising the Kullback-Leibler divergence.
  In practice this step is obtained by a fixed-point algorithm.

- **M-Step** maximizes $\mathcal{J}(\theta, \tau)$ with respect to θ:

$$\widehat{\theta}^{(t)} = \arg\max_{\theta} \mathcal{J}(\theta, \tau^{(t-1)}), \quad \text{s.t. } \sum_{q=1}^{Q} \pi_q = 1,$$

  thus updating the value of the model parameters π_q and $B^{(m)}_{q_1,\ldots,q_m}$.

Model selection and generalizations

Integrated classification likelihood (ICL)

We select $\hat{q} = \arg\max_q ICL(q)$ where

$$\mathrm{ICL}(q) = \log \mathbb{P}_{\hat{\theta}}(\boldsymbol{Y}, \hat{\boldsymbol{Z}}) - \frac{1}{2}(q-1)\log n - \frac{1}{2}\sum_{m=2}^{M} \binom{q+m-1}{m} \log \binom{n}{m}.$$

Generalizations

- We have not considered self-loops ($m = 1$) but it's easy to do;
- Binary hyperedge variables could be replaced by counting hyperedges variables, replacing the Bernoulli distribution with, for e.g. (zero-inflated or deflated) Poisson law.

Computational complexity - and considerations on the choice of M

- Focusing on *simple* hypergraphs **has a high price**: we need to explore all the $\binom{n}{m}$ tuples of nodes for all $2 \leq m \leq M$;
- Our algorithm has a complexity of $O(n\binom{n}{M}Q^M)$, which is huge;
- Current modularity approaches avoid this issue by working with multisets-hypergraphs, because there the summations over multisets of nodes $\sum_{i_1,\dots,i_m}$ factorize into m independent sums (no constraint that the nodes be different), and this further simplifies the expression of the modularity;
- Again, this is inappropriate on some datasets;
- As a consequence: we recommend to use **a reasonable value of M**: indeed M is not necessarily the largest observed hyperedge size (e.g. co-authorship dataset);
  $\hookrightarrow$ base your clustering on size-m hyperedges with $m \leq M$
  $\hookrightarrow$ means you don't use HOI of size larger than M but you still do better than with clique-expansion graphs!

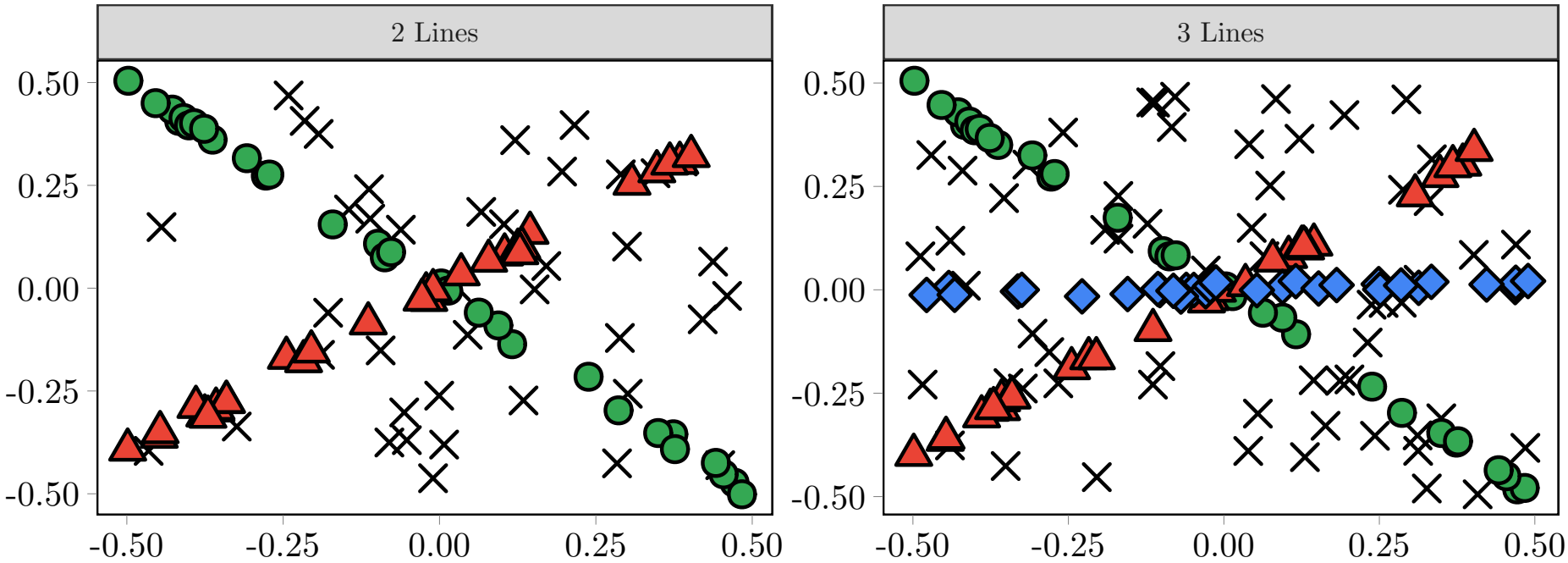
Outline

Simulations

We've done simulations under the model (and in a sparse setting): it works well (trust me, or look at the paper).

Line clustering through hypergraphs I

2 experiments: 2 lines (3 groups) and 3 lines (4 groups)



Line clustering through hypergraphs II

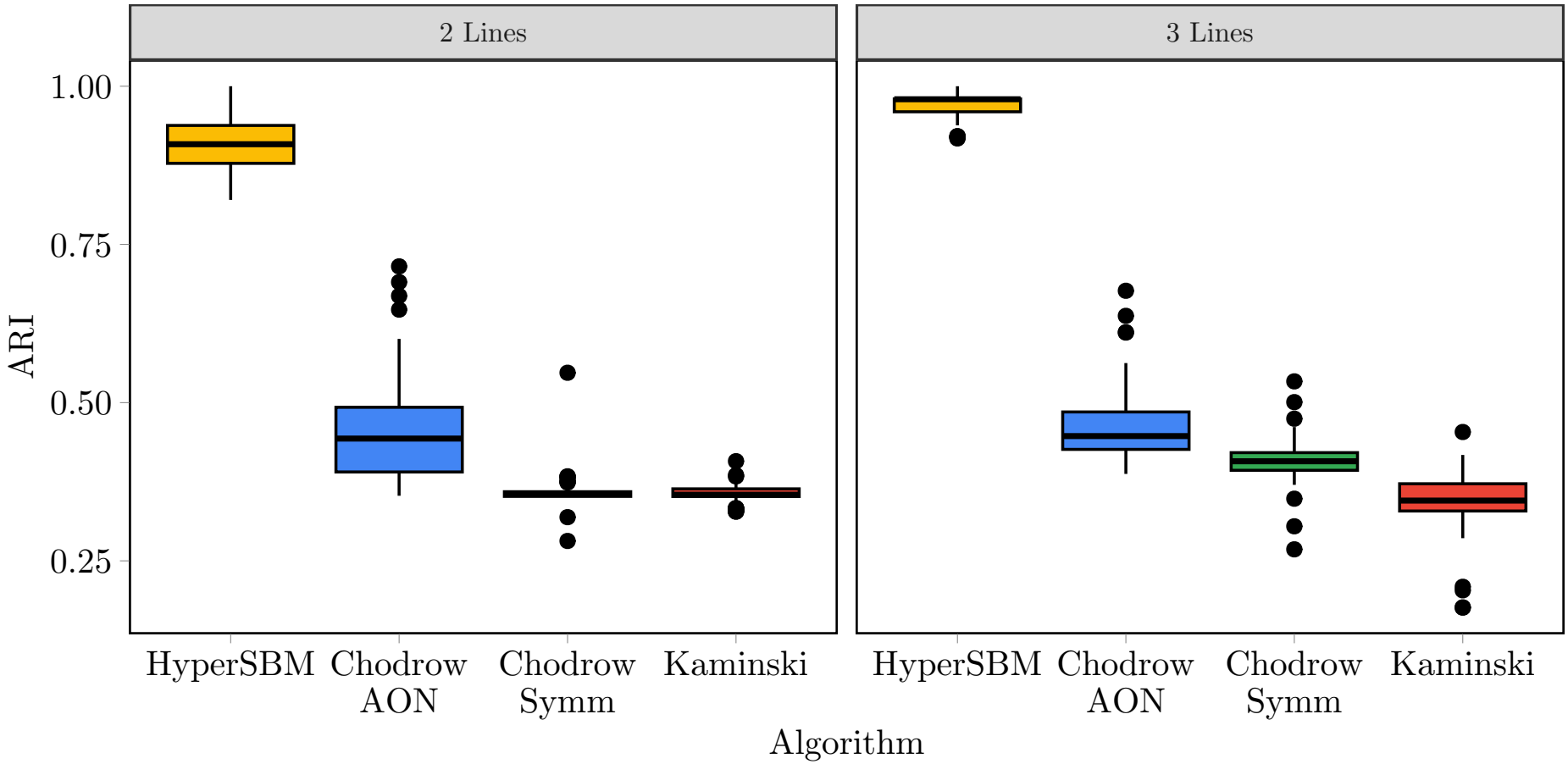
Hypergraph construction

- Select 3 points at random and fit a line
- If residual distance is less than a threshold, draw a hyperedge between those 3 points
- Globally set signal:noise hyperedge ratio = 2
- Repeat to obtain 100 3-uniform hypergraphs

Data characteristics

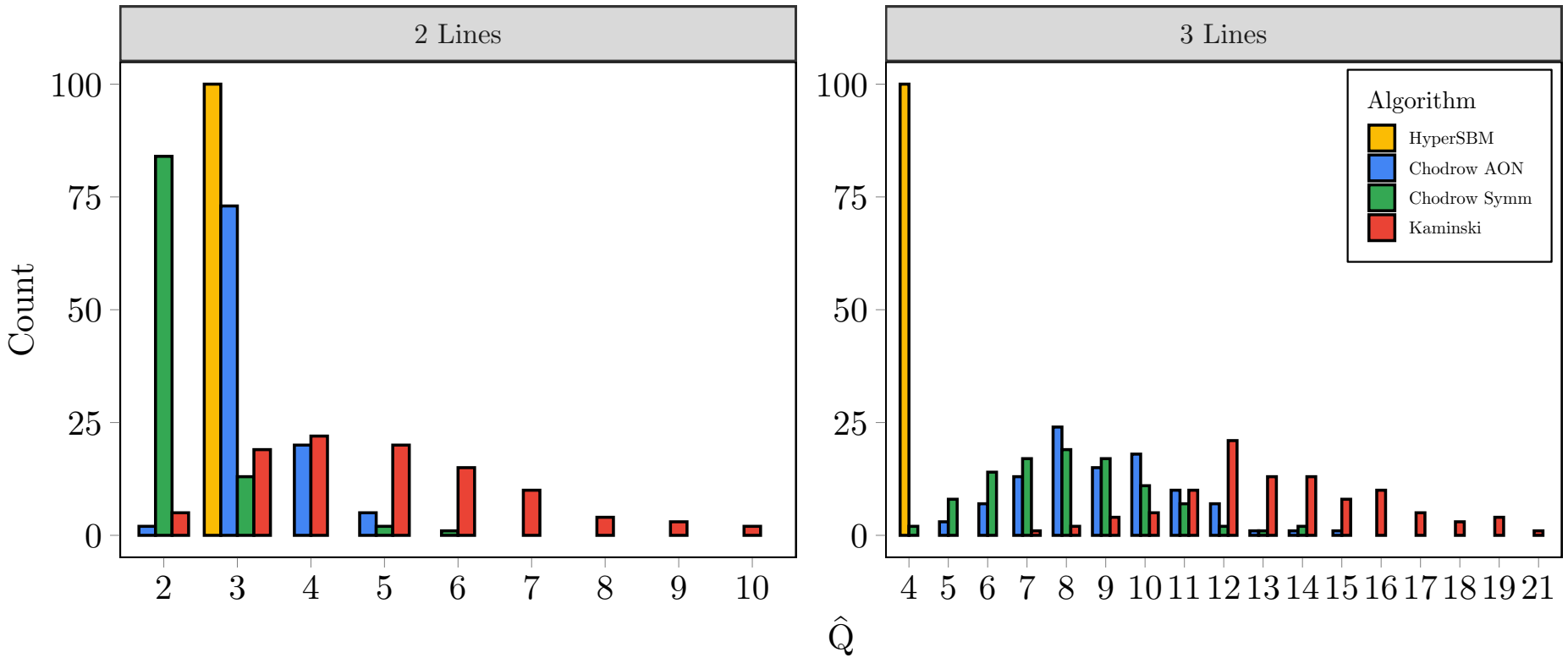
	Pts/line	Noisy pts	Total nb pts	mean nb of hyperedges
2 lines	30	40	100	1070.84
3 lines	30	60	150	587.7

Comparison with modularity based methods I



Adjusted Rand Index

Comparison with modularity based methods II



Estimated number of groups

Outline

Conclusions

- Higher-order interactions is the new trend;
- There are already some available tools that you can test on your datasets;
  - $\hookrightarrow$ do you have such **new** datasets?
  - $\hookrightarrow$ may want to re-visit all bipartites graphs (constructing 2 different hypergraphs for each of these!)
- $\hookrightarrow$ at the moment, there is a lack of large scale characteristics of hypergraphs;
- Among the many open questions: detectability limits for non-uniform hypergraphs.

Any questions?

References I

Battiston, F., G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, and G. Petri (2020). Networks beyond pairwise interactions: Structure and dynamics. *Physics Reports 874*, 1–92.

Brusa, L. and C. Matias (2022). Model-based clustering in simple hypergraphs through a stochastic blockmodel. Technical report, arXiv:2210.05983.

Chodrow, P. S., N. Veldt, and A. R. Benson (2021). Generative hypergraph clustering: From blockmodels to modularity. *Science Advances 7*(28), eabh1303.

Ghoshdastidar, D. and A. Dukkipati (2014). Consistency of spectral partitioning of uniform hypergraphs under planted partition model. In *Advances in Neural Information Processing Systems*, Volume 27.

Ghoshdastidar, D. and A. Dukkipati (2017). Consistency of spectral hypergraph partitioning under planted partition model. *The Annals of Statistics 45*(1), 289 – 315.

Kamiński, B., V. Poulin, P. Prałat, P. Szufel, and F. Théberge (2019a). Clustering via hypergraph modularity. *PLoS ONE 14*(11), e0224307.

References II

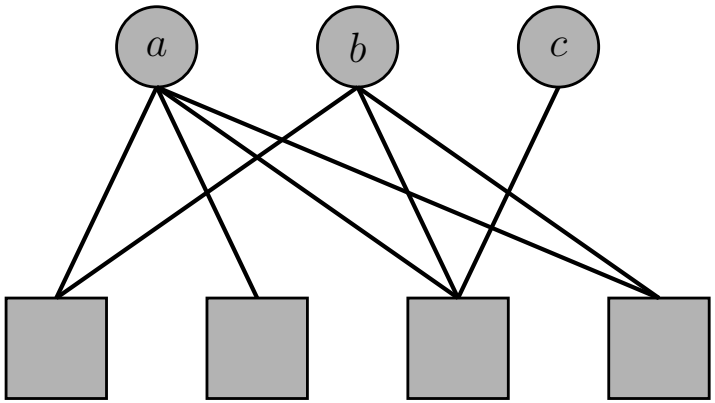
Kumar, T., S. Vaidyanathan, H. Ananthapadmanabhan, S. Parthasarathy, and B. Ravindran (2020). Hypergraph clustering by iteratively reweighted modularity maximization. *Appl. Netw. Sci. 5*(1), 52.

M. T. Schaub *et al.* (2021). Signal processing on higher-order networks: Livin' on the edge... and beyond. /em Signal processing, Volume 187, 108149

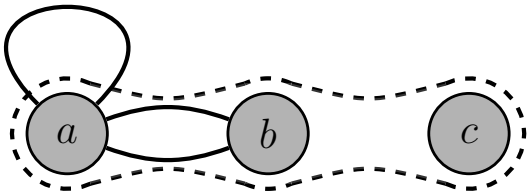
Non equivalence between simple binary hypergraphs and bipartite graphs

Bipartite graphs space

Hypergraphs space



(a)



(b)