Graph clustering and ranking for lead-lag detection in equity markets

Mihai Cucuringu

University of Oxford
Oxford-Man Institute (OMI)
The Alan Turing Institute

Oxford Machine Learning Summer School
9 July 2023

Signed graph clustering

Financial time series clustering & cluster portfolios

Directed graph clustering

Lead-lag detection in financial multivariate time series

Overview
MetaCluster Lead-lag Portfolios
GlobalRank Lead-lag Portfolios
ClusterRank Lead-lag Portfolios

Main motivation: build a network from time series data, leverage structural properties of the network to inform downstream time series tasks (eg prediction).

Network of financial assets

- Mel MacMahon and Diego Garlaschelli. Phys.Rev.X5, 2015. *Community Detection for Correlation Matrices*.

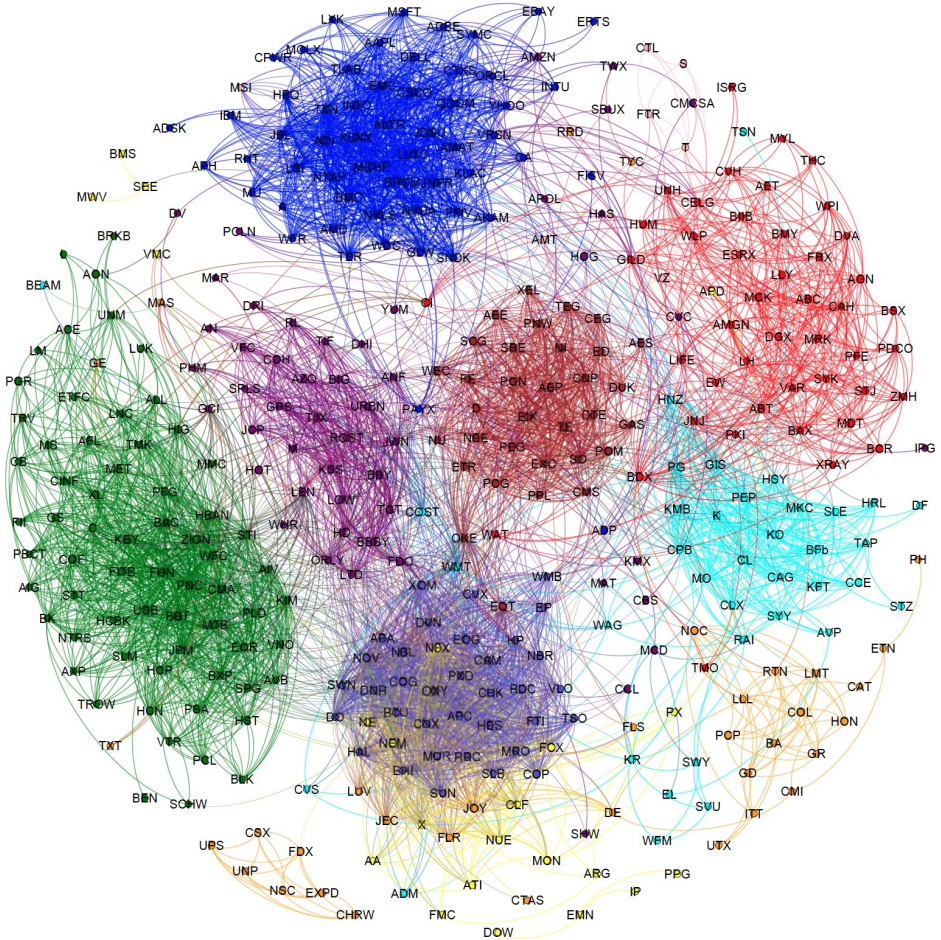


Figure: Asset correlation matrix after thresholding. The color of each node represents the industry sector to which that stock belongs. The force-based layout clearly indicates the existence of strong connections between stocks of the same industry sector.

Clustering graphs

- Consider an undirected graph $G = (V, E)$ with n vertices
- Each $\{i, j\} \in E$ has an associated **positive** weight $w_{ij} > 0$ (similarity between vertices)
- Potentially constructed via a kernel K_ϵ such that

$$w_{ij} = K_\epsilon(||x_i - x_j||), \quad (ij) \in E(G) \tag{1}$$

Clustering graphs

- Consider an undirected graph $G = (V, E)$ with n vertices
- Each $\{i, j\} \in E$ has an associated **positive** weight $w_{ij} > 0$ (similarity between vertices)
- Potentially constructed via a kernel K_ϵ such that

$$w_{ij} = K_\epsilon(||x_i - x_j||), \quad (ij) \in E(G) \tag{1}$$

- **Goal:** Partition V into **clusters** s.t. intra-cluster edges have high weight and inter-cluster edges have low weight.

Clustering graphs

- Consider an undirected graph $G = (V, E)$ with n vertices
- Each $\{i, j\} \in E$ has an associated **positive** weight $w_{ij} > 0$ (similarity between vertices)
- Potentially constructed via a kernel K_ϵ such that

$$w_{ij} = K_\epsilon(||x_i - x_j||), \quad (ij) \in E(G) \tag{1}$$

- **Goal:** Partition V into **clusters** s.t. intra-cluster edges have high weight and inter-cluster edges have low weight.
- **Applications:** Statistics, computer science, biology etc.

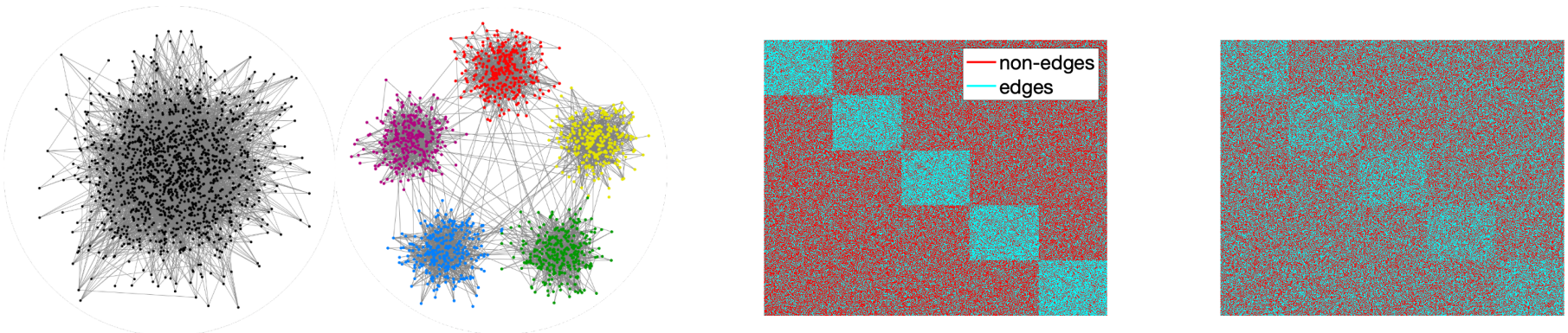


Figure: [Abbe '17] Recover the right graph from the left (scrambled) graph.

Clustering graphs

- Consider an undirected graph $G = (V, E)$ with n vertices
- Each $\{i, j\} \in E$ has an associated **positive** weight $w_{ij} > 0$ (similarity between vertices)
- Potentially constructed via a kernel K_ϵ such that

$$w_{ij} = K_\epsilon(||x_i - x_j||), \quad (ij) \in E(G) \tag{1}$$

- **Goal:** Partition V into **clusters** s.t. intra-cluster edges have high weight and inter-cluster edges have low weight.
- **Applications:** Statistics, computer science, biology etc.

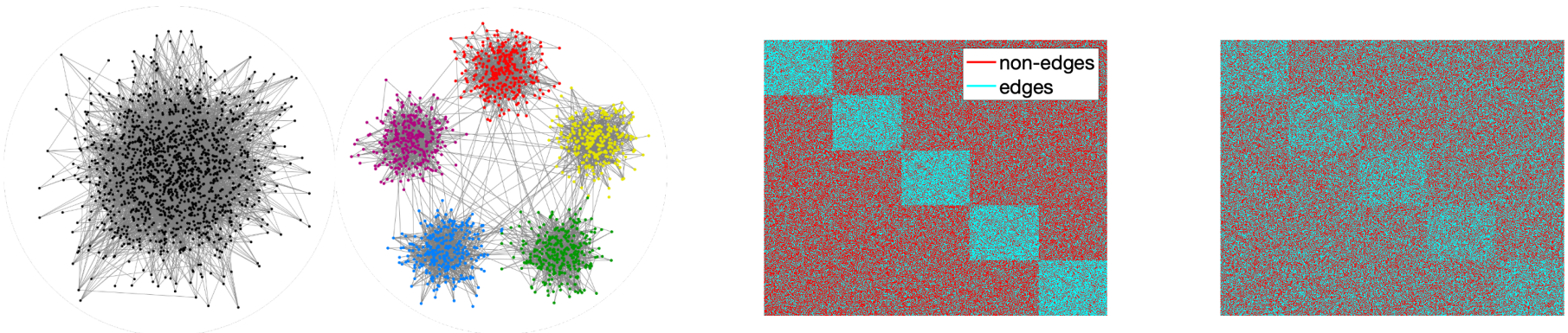


Figure: [Abbe '17] Recover the right graph from the left (scrambled) graph.

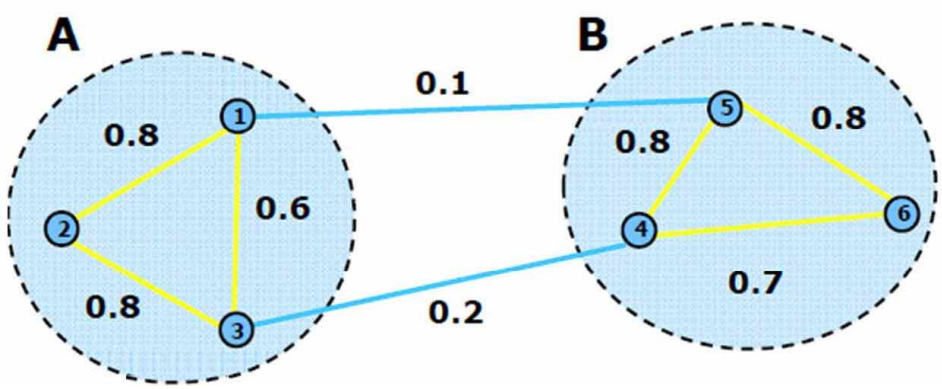
Unlike standard graph clustering settings, we do not require that the intra-cluster edge probabilities to be different from those of inter-cluster edges

- implicitly achieved by the sign or directionality of the edges

Graph Cuts

- consider a partition of the graph G into two subgraphs A and B
- the Cut(A, B) will be given by the sum of the weights of the set of edges that connect the two groups

$$\text{cut(A,B)} := \sum_{i \in A, j \in B} w_{ij} \tag{2}$$



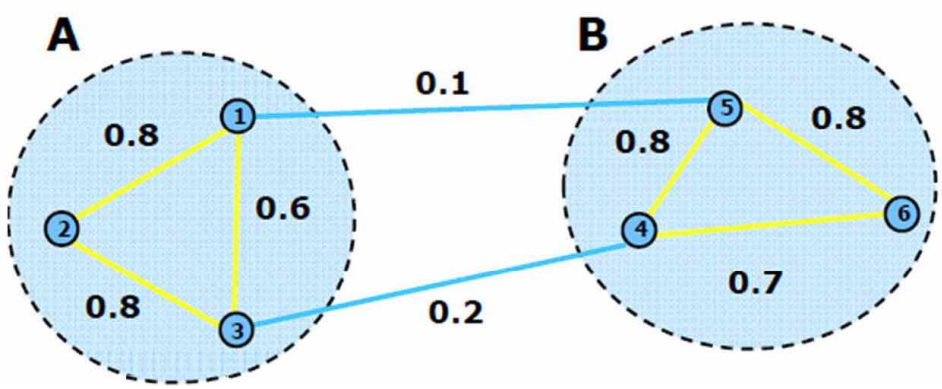
[Source: David Sontag]

- Cut(A) = Cut$(A, \overline{A})$

Graph Cuts

- consider a partition of the graph G into two subgraphs A and B
- the Cut(A, B) will be given by the sum of the weights of the set of edges that connect the two groups

$$\text{cut(A,B)} := \sum_{i \in A, j \in B} w_{ij} \tag{2}$$



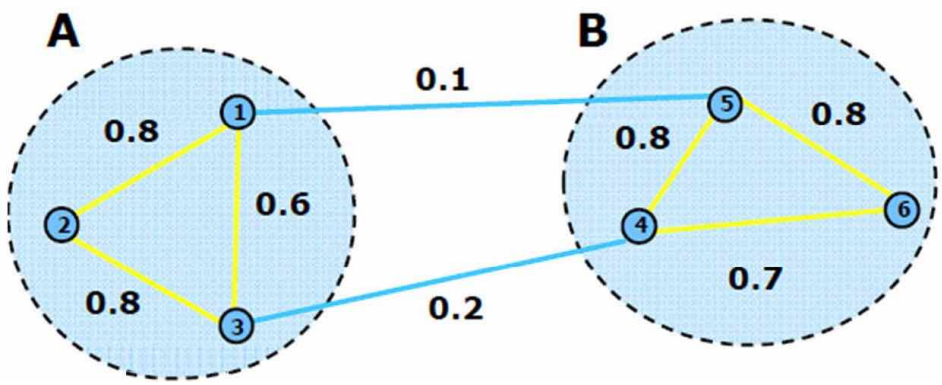
[Source: David Sontag]

- Cut(A) = Cut$(A, \overline{A})$
- the notion of a **cut** is a fundamental concept in graph clustering

Graph Cuts

- consider a partition of the graph G into two subgraphs A and B
- the Cut(A, B) will be given by the sum of the weights of the set of edges that connect the two groups

$$\text{cut(A,B)} := \sum_{i \in A, j \in B} w_{ij} \tag{2}$$



[Source: David Sontag]

- Cut(A) = Cut$(A, \overline{A})$
- the notion of a **cut** is a fundamental concept in graph clustering
- aim to find a partition/split of G into A and B in order to minimize the resulting cut.

Recall the definition of the graph Laplacian

- Graph Laplacian $L = D - A$ (most popular version)

Recall the definition of the graph Laplacian

- Graph Laplacian $L = D - A$ (most popular version)
- A is the adjacency matrix of the graph $A_{ij} \geq 0$

Recall the definition of the graph Laplacian

- Graph Laplacian $L = D - A$ (most popular version)
- A is the adjacency matrix of the graph $A_{ij} \geq 0$
- D is a diagonal matrix, D_{ii} denoting the degree of node i

Recall the definition of the graph Laplacian

- Graph Laplacian $L = D - A$ (most popular version)
- A is the adjacency matrix of the graph $A_{ij} \geq 0$
- D is a diagonal matrix, D_{ii} denoting the degree of node i

$$L(i,j) \stackrel{\text{def}}{=} \begin{cases} deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \quad \text{and} \quad (i,j) \in E(G) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Recall the definition of the graph Laplacian

- Graph Laplacian $L = D - A$ (most popular version)
- A is the adjacency matrix of the graph $A_{ij} \geq 0$
- D is a diagonal matrix, D_{ii} denoting the degree of node i

$$L(i,j) \stackrel{\text{def}}{=} \begin{cases} deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \quad \text{and} \quad (i,j) \in E(G) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

- L is symmetric

Recall the definition of the graph Laplacian

- Graph Laplacian $L = D - A$ (most popular version)
- A is the adjacency matrix of the graph $A_{ij} \geq 0$
- D is a diagonal matrix, D_{ii} denoting the degree of node i

$$L(i,j) \stackrel{\text{def}}{=} \begin{cases} deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \quad \text{and} \quad (i,j) \in E(G) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

- L is symmetric
- eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_{n-1}$, eigenvectors $\phi_0, \phi_1, \ldots, \phi_{n-1}$

Recall the definition of the graph Laplacian

- Graph Laplacian $L = D - A$ (most popular version)
- A is the adjacency matrix of the graph $A_{ij} \geq 0$
- D is a diagonal matrix, D_{ii} denoting the degree of node i

$$L(i,j) \stackrel{\text{def}}{=} \begin{cases} deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \quad \text{and} \quad (i,j) \in E(G) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

- L is symmetric
- eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_{n-1}$, eigenvectors $\phi_0, \phi_1, \ldots, \phi_{n-1}$
- every row sum and column sum of L is zero

Recall the definition of the graph Laplacian

- Graph Laplacian $L = D - A$ (most popular version)
- A is the adjacency matrix of the graph $A_{ij} \geq 0$
- D is a diagonal matrix, D_{ii} denoting the degree of node i

$$L(i,j) \stackrel{\text{def}}{=} \begin{cases} deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \quad \text{and} \quad (i,j) \in E(G) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

- L is symmetric
- eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_{n-1}$, eigenvectors $\phi_0, \phi_1, \ldots, \phi_{n-1}$
- every row sum and column sum of L is zero
- thus, $\lambda_0 = 0$, and $\phi_0 = \mathbf{1} \stackrel{\text{def}}{=} [1, 1, \ldots, 1]^T$ since $L\,\mathbf{1} = 0\,\mathbf{1}$

Recall the definition of the graph Laplacian

- Graph Laplacian $L = D - A$ (most popular version)
- A is the adjacency matrix of the graph $A_{ij} \geq 0$
- D is a diagonal matrix, D_{ii} denoting the degree of node i

$$L(i,j) \stackrel{\text{def}}{=} \begin{cases} deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \quad \text{and} \quad (i,j) \in E(G) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

- L is symmetric
- eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_{n-1}$, eigenvectors $\phi_0, \phi_1, \ldots, \phi_{n-1}$
- every row sum and column sum of L is zero
- thus, $\lambda_0 = 0$, and $\phi_0 = \mathbf{1} \stackrel{\text{def}}{=} [1, 1, \ldots, 1]^T$ since $L\,\mathbf{1} = 0\,\mathbf{1}$
- the second smallest (smallest non-zero) eigenvalue of L is the algebraic connectivity (Fiedler value, spectral gap) of G

Recall the definition of the graph Laplacian

- Graph Laplacian $L = D - A$ (most popular version)
- A is the adjacency matrix of the graph $A_{ij} \geq 0$
- D is a diagonal matrix, D_{ii} denoting the degree of node i

$$L(i,j) \stackrel{\text{def}}{=} \begin{cases} deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \quad \text{and} \quad (i,j) \in E(G) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

- L is symmetric
- eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_{n-1}$, eigenvectors $\phi_0, \phi_1, \ldots, \phi_{n-1}$
- every row sum and column sum of L is zero
- thus, $\lambda_0 = 0$, and $\phi_0 = \mathbf{1} \stackrel{\text{def}}{=} [1, 1, \ldots, 1]^T$ since $L\,\mathbf{1} = 0\,\mathbf{1}$
- the second smallest (smallest non-zero) eigenvalue of L is the algebraic connectivity (Fiedler value, spectral gap) of G

Lemma If $G = (V, E)$ is **connected** and $\lambda_0 \leq \lambda_1 \leq \lambda_{n-1}$ are the eigenvalues of its Laplacian L, then it holds true that $\lambda_1 > 0$.

Recall the definition of the graph Laplacian

- Graph Laplacian $L = D - A$ (most popular version)
- A is the adjacency matrix of the graph $A_{ij} \geq 0$
- D is a diagonal matrix, D_{ii} denoting the degree of node i

$$L(i,j) \stackrel{\text{def}}{=} \begin{cases} deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \quad \text{and} \quad (i,j) \in E(G) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

- L is symmetric
- eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_{n-1}$, eigenvectors $\phi_0, \phi_1, \ldots, \phi_{n-1}$
- every row sum and column sum of L is zero
- thus, $\lambda_0 = 0$, and $\phi_0 = \mathbf{1} \stackrel{\text{def}}{=} [1, 1, \ldots, 1]^T$ since $L\,\mathbf{1} = 0\,\mathbf{1}$
- the second smallest (smallest non-zero) eigenvalue of L is the algebraic connectivity (Fiedler value, spectral gap) of G

Lemma If $G = (V, E)$ is **connected** and $\lambda_0 \leq \lambda_1 \leq \lambda_{n-1}$ are the eigenvalues of its Laplacian L, then it holds true that $\lambda_1 > 0$. (Stronger result: the multiplicity of the zero eigenvalue is equal to the number of connected components).

Spectral clustering of (unsigned) graphs

- A popular approach is to perform spectral clustering:
  - **Idea:** Embed V into $\mathbb{R}^k$ and perform k means clustering.
  - Embedding obtained from extremal eigenvectors of suitable graph matrix (eg., Laplacian).

Spectral clustering of (unsigned) graphs

- A popular approach is to perform spectral clustering:
  - **Idea:** Embed V into $\mathbb{R}^k$ and perform k means clustering.
  - Embedding obtained from extremal eigenvectors of suitable graph matrix (eg., Laplacian).
- **Example:** Normalized cut (NC) [Shi and Malik, 2000]

$$\min_{C_1,\ldots,C_k} \sum_{i=1,\ldots,k} \frac{\mathrm{cut}(C_i)}{\mathrm{vol}(C_i)}$$

  - $\mathrm{cut}(A) := \sum_{i\in A, j\notin A} w_{ij}$ and $\mathrm{vol}(A) := \sum_{v\in A} \deg(v)$

Spectral clustering of (unsigned) graphs

- A popular approach is to perform spectral clustering:
  - **Idea:** Embed V into $\mathbb{R}^k$ and perform k means clustering.
  - Embedding obtained from extremal eigenvectors of suitable graph matrix (eg., Laplacian).
- **Example:** Normalized cut (NC) [Shi and Malik, 2000]

$$\min_{C_1,\ldots,C_k} \sum_{i=1,\ldots,k} \frac{\mathrm{cut}(C_i)}{\mathrm{vol}(C_i)}$$

  - $\mathrm{cut}(A) := \sum_{i\in A, j\notin A} w_{ij}$ and $\mathrm{vol}(A) := \sum_{v\in A} \deg(v)$
- NC is a discrete optimization problem, NP-hard in worst case.
- We can "**relax**" the discrete constraints in NC: the solution of new problem is given by the smallest k eigenvectors of

$$D^{-1/2} L D^{-1/2}$$

Spectral clustering of (unsigned) graphs

- A popular approach is to perform spectral clustering:
  - **Idea:** Embed V into $\mathbb{R}^k$ and perform k means clustering.
  - Embedding obtained from extremal eigenvectors of suitable graph matrix (eg., Laplacian).
- **Example:** Normalized cut (NC) [Shi and Malik, 2000]

$$\min_{C_1,\ldots,C_k} \sum_{i=1,\ldots,k} \frac{\mathrm{cut}(C_i)}{\mathrm{vol}(C_i)}$$

  - $\mathrm{cut}(A) := \sum_{i\in A, j\notin A} w_{ij}$ and $\mathrm{vol}(A) := \sum_{v\in A} \deg(v)$
- NC is a discrete optimization problem, NP-hard in worst case.
- We can "**relax**" the discrete constraints in NC: the solution of new problem is given by the smallest k eigenvectors of

$$D^{-1/2} L D^{-1/2}$$

  - D: diagonal matrix with the degrees
  - $L = D - A$: the Laplacian of G

Spectral clustering of (unsigned) graphs

- A popular approach is to perform spectral clustering:
  - **Idea:** Embed V into $\mathbb{R}^k$ and perform k means clustering.
  - Embedding obtained from extremal eigenvectors of suitable graph matrix (eg., Laplacian).
- **Example:** Normalized cut (NC) [Shi and Malik, 2000]

$$\min_{C_1,\ldots,C_k} \sum_{i=1,\ldots,k} \frac{\mathrm{cut}(C_i)}{\mathrm{vol}(C_i)}$$

  - $\mathrm{cut}(A) := \sum_{i\in A, j\notin A} w_{ij}$ and $\mathrm{vol}(A) := \sum_{v\in A} \deg(v)$
- NC is a discrete optimization problem, NP-hard in worst case.
- We can "**relax**" the discrete constraints in NC: the solution of new problem is given by the smallest k eigenvectors of

$$D^{-1/2} L D^{-1/2}$$

  - D: diagonal matrix with the degrees
  - $L = D - A$: the Laplacian of G
- $n \times k$ eigenvector matrix is the graph embedding in R^k

Spectral clustering of (unsigned) graphs

- A popular approach is to perform spectral clustering:
  - **Idea:** Embed V into $\mathbb{R}^k$ and perform k means clustering.
  - Embedding obtained from extremal eigenvectors of suitable graph matrix (eg., Laplacian).
- **Example:** Normalized cut (NC) [Shi and Malik, 2000]

$$\min_{C_1,\ldots,C_k} \sum_{i=1,\ldots,k} \frac{\mathrm{cut}(C_i)}{\mathrm{vol}(C_i)}$$

  - $\mathrm{cut}(A) := \sum_{i\in A, j\notin A} w_{ij}$ and $\mathrm{vol}(A) := \sum_{v\in A} \deg(v)$
- NC is a discrete optimization problem, NP-hard in worst case.
- We can "**relax**" the discrete constraints in NC: the solution of new problem is given by the smallest k eigenvectors of

$$D^{-1/2} L D^{-1/2}$$

  - D: diagonal matrix with the degrees
  - $L = D - A$: the Laplacian of G
- $n \times k$ eigenvector matrix is the graph embedding in R^k

---

Laplacian $L = D - A$ is PSD; captures the cut if x is a cluster indicator vector

$$x^T L\, x = \sum_{(i,j)\in E} (x_i - x_j)^2 = \text{total edge weight leaving the cluster}$$

Clustering (unsigned) graphs in a financial context

Hamed Amini, Yudong Chen, Andreea Minca, Xin Qian

Clustering Heterogeneous Financial Networks

Mathematical Finance (2023)

- considers the problem of clustering algorithms for financial networks, with an eye towards heterogeneity

Clustering (unsigned) graphs in a financial context

Hamed Amini, Yudong Chen, Andreea Minca, Xin Qian

Clustering Heterogeneous Financial Networks

Mathematical Finance (2023)

- considers the problem of clustering algorithms for financial networks, with an eye towards heterogeneity
- provide applications to systemic risk and portfolio diversification

Clustering (unsigned) graphs in a financial context

Hamed Amini, Yudong Chen, Andreea Minca, Xin Qian

Clustering Heterogeneous Financial Networks

Mathematical Finance (2023)

- considers the problem of clustering algorithms for financial networks, with an eye towards heterogeneity
- provide applications to systemic risk and portfolio diversification
- introduced a semidefinite programming (SDP)-based clustering algorithm with a suitability chosen regularization term aimed at handling outliers

Clustering (unsigned) graphs in a financial context

Hamed Amini, Yudong Chen, Andreea Minca, Xin Qian

Clustering Heterogeneous Financial Networks

Mathematical Finance (2023)

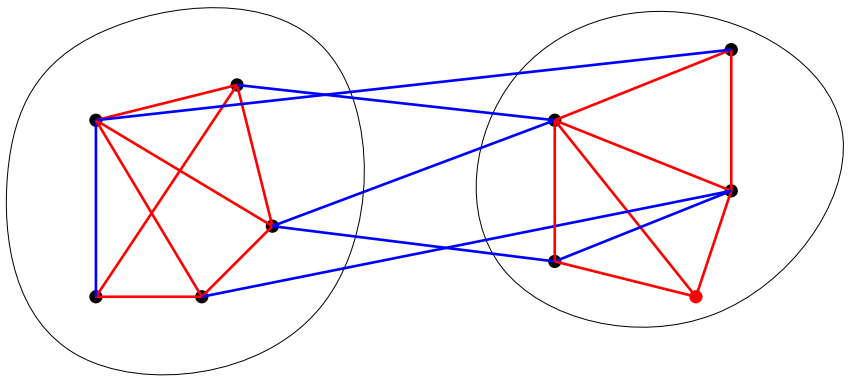
- considers the problem of clustering algorithms for financial networks, with an eye towards heterogeneity
- provide applications to systemic risk and portfolio diversification
- introduced a semidefinite programming (SDP)-based clustering algorithm with a suitability chosen regularization term aimed at handling outliers
- theoretical analysis provides exact recovery guarantees of the inlier nodes with high probability

Signed Graphs and Signed Graph Cuts

- Many applications involve graphs where edge weights can take negative values (dissimilarity) as well
  - **Social networks:** users are friends (“+" edge) or enemies (“–" edge)
  - **Image segmentation:** edges connect adjacent pixels and encode (dis-)similarity information

Signed Graphs and Signed Graph Cuts

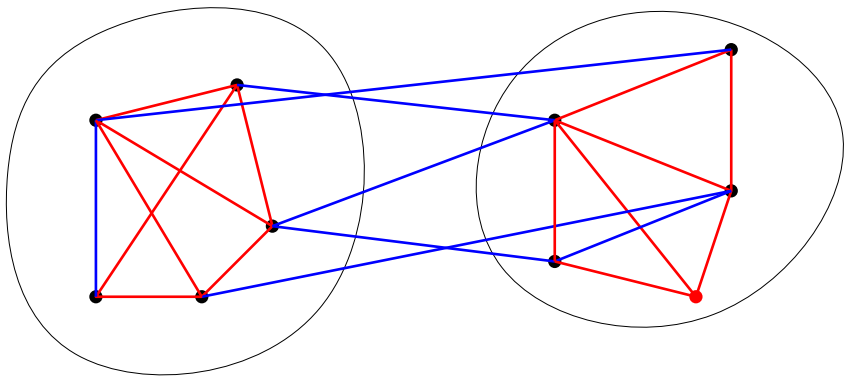
- Many applications involve graphs where edge weights can take negative values (dissimilarity) as well
  - **Social networks:** users are friends (“+" edge) or enemies (“–" edge)
  - **Image segmentation:** edges connect adjacent pixels and encode (dis-)similarity information
  - **Time series clustering**: correlation clustering



Blue edge: $-$ Red edge: $+$

Signed Graphs and Signed Graph Cuts

- Many applications involve graphs where edge weights can take negative values (dissimilarity) as well
  - **Social networks:** users are friends ("+" edge) or enemies ("–" edge)
  - **Image segmentation:** edges connect adjacent pixels and encode (dis-)similarity information
  - **Time series clustering**: correlation clustering



Blue edge: $-$ Red edge: $+$

Goal:
Maximize the sum of weights of: intra cluster positive edges plus inter cluster negative edges

Motivation from Statistical Arbitrage

Stat arb broadly refers to

- technical short-term mean-reversion strategies
- involving a large numbers of financial instruments (hundreds to thousands)
- very short holding periods (days to seconds)
- significant computational, trading, and technology infrastructure

Motivation from Statistical Arbitrage

Stat arb broadly refers to

- technical short-term mean-reversion strategies
- involving a large numbers of financial instruments (hundreds to thousands)
- very short holding periods (days to seconds)
- significant computational, trading, and technology infrastructure

Basic idea behind certain types of statistical arbitrage trading strategies:

- certain quantities are historically correlated
- occasionally these correlations are temporarily undone by certain unusual market conditions
- one expects the previous correlations will be restored in the future

Motivation from Statistical Arbitrage

Stat arb broadly refers to

- technical short-term mean-reversion strategies
- involving a large numbers of financial instruments (hundreds to thousands)
- very short holding periods (days to seconds)
- significant computational, trading, and technology infrastructure

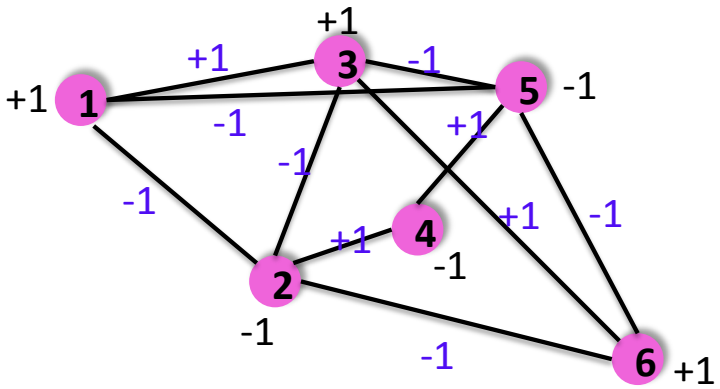
Basic idea behind certain types of statistical arbitrage trading strategies:

- certain quantities are historically correlated
- occasionally these correlations are temporarily undone by certain unusual market conditions
- one expects the previous correlations will be restored in the future

Pairs-trading - widely assumed to be the *ancestor* of statistical arbitrage:

- If stocks X and Y are in the same industry or have similar characteristics (e.g. Pepsi and Coca Cola), one expects the returns of the two stocks to track each other after controlling for beta.
- cluster of size 2

Signed clustering with $k = 2$ (group synchronization over $\mathbb{Z}_2$)



Signed clustering with $k = 2$ (group synchronization over $\mathbb{Z}_2$)

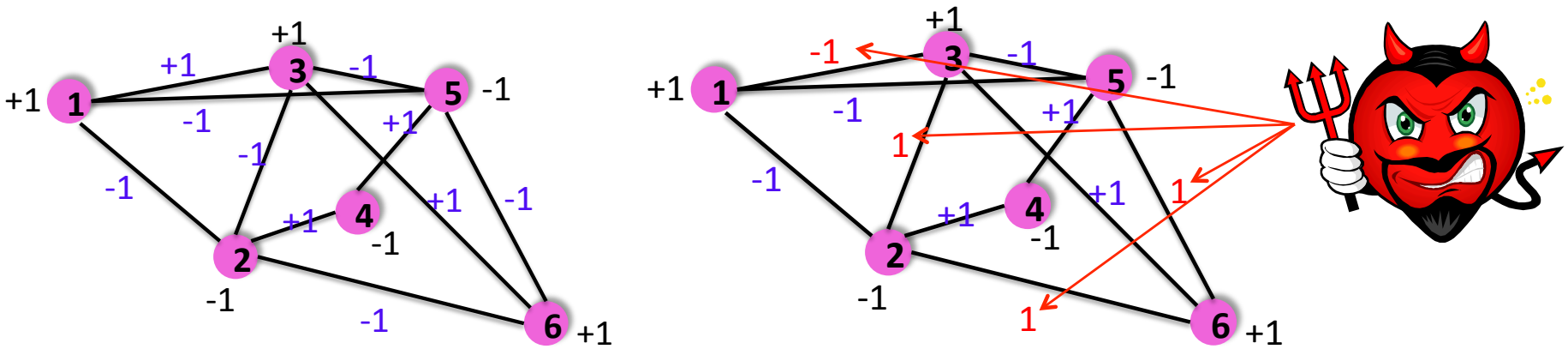


Figure: Synchronization over $\mathbb{Z}_2$ (left: clean, right: noisy)

- unknown group elements $z_1, z_2, \ldots, z_N \in \mathbb{Z}_2$ (eg. ± 1) correspond to the vertices of a measurement graph G
- $z_i z_j$ encodes the measured similarity between nodes (eg., stocks) i and j

Signed clustering with $k = 2$ (group synchronization over $\mathbb{Z}_2$)

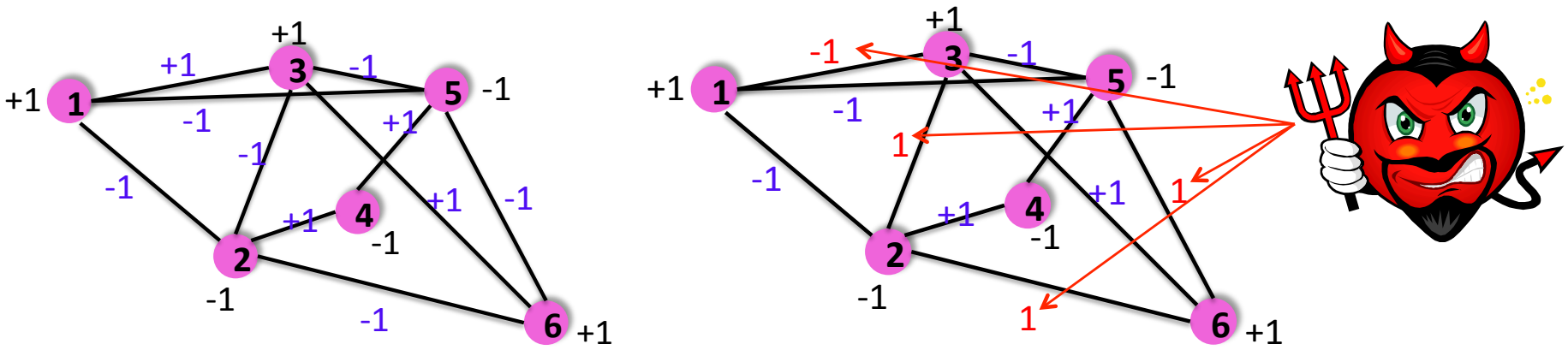


Figure: Synchronization over $\mathbb{Z}_2$ (left: clean, right: noisy)

- unknown group elements $z_1, z_2, \ldots, z_N \in \mathbb{Z}_2$ (eg. ± 1) correspond to the vertices of a measurement graph G
- $z_i z_j$ encodes the measured similarity between nodes (eg., stocks) i and j
- a potential noise model for the measurement graph is

$$A_{ij} = \begin{cases} z_i z_j & (i,j) \in E \text{ and the measurement is correct,} \\ -z_i z_j & (i,j) \in E \text{ and the measurement is incorrect,} \\ 0 & (i,j) \notin E \end{cases}$$

Signed clustering with $k = 2$ (group synchronization over $\mathbb{Z}_2$)

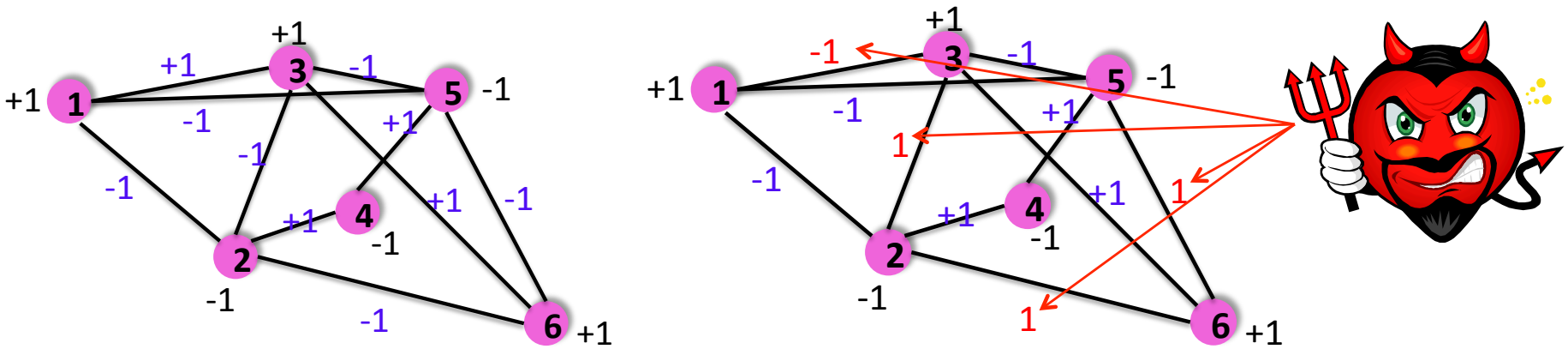


Figure: Synchronization over $\mathbb{Z}_2$ (left: clean, right: noisy)

- unknown group elements $z_1, z_2, \ldots, z_N \in \mathbb{Z}_2$ (eg. ± 1) correspond to the vertices of a measurement graph G
- $z_i z_j$ encodes the measured similarity between nodes (eg., stocks) i and j
- a potential noise model for the measurement graph is

$$A_{ij} = \begin{cases} z_i z_j & (i,j) \in E \text{ and the measurement is correct,} \\ -z_i z_j & (i,j) \in E \text{ and the measurement is incorrect,} \\ 0 & (i,j) \notin E \end{cases}$$

- original solution: $z_1, \ldots, z_n \in \pm 1^n \quad (\mathbb{Z}_2 = \{-1, +1\})$
- task: estimate approx. solution $x_1, \ldots, x_N \in \pm 1^N$ such that we satisfy as many pairwise group relations (happy edges) in $\mathbb{Z}_2$ as possible.

Signed clustering with k=2 (Synchronization over $\mathbb{Z}_2$)

- Consider maximizing the following quadratic form (intra-cluster happiness)

$$\max_{x_1,\ldots,x_N \in \mathbb{Z}_2^N} \sum_{i,j=1}^{N} x_i A_{ij} x_j = \max_{x_1,\ldots,x_N \in \mathbb{Z}_2^N} x^T A x, \tag{4}$$

whose maximum is attained at $x = z$ (noise-free data). NP-hard, but relax to

$$\max_{\sum_{i=1}^{N} |x_i|^2 = N} \sum_{i,j=1}^{N} x_i A_{ij} x_j = \max_{\|x\|^2 = N} x^T A x \tag{5}$$

whose max is achieved when $x = v_1$, the normalized top eigenvector of A

$$A v_1 = \lambda_1 v_1$$

Signed clustering with k=2 (Synchronization over $\mathbb{Z}_2$)

- Consider maximizing the following quadratic form (intra-cluster happiness)

$$\max_{x_1,\ldots,x_N \in \mathbb{Z}_2^N} \sum_{i,j=1}^{N} x_i A_{ij} x_j = \max_{x_1,\ldots,x_N \in \mathbb{Z}_2^N} x^T A x, \tag{4}$$

whose maximum is attained at $x = z$ (noise-free data). NP-hard, but relax to

$$\max_{\sum_{i=1}^N |x_i|^2 = N} \sum_{i,j=1}^{N} x_i A_{ij} x_j = \max_{\|x\|^2 = N} x^T A x \tag{5}$$

whose max is achieved when $x = v_1$, the normalized top eigenvector of A

$$A v_1 = \lambda_1 v_1$$

Alternatively, formulate synchronization as a least squares problem, by minimizing the following quadratic form (minimize unhappy edges)

$$\min_{x \in \mathbb{Z}_2^N} \sum_{(i,j) \in E} (x_i - A_{ij} x_j)^2 = \ldots = \min_{x \in \mathbb{Z}_2^N} x^T (\overline{D} - A) x$$

- (psd) Signed Laplacian $\overline{L} = \overline{D} - A$, diagonal matrix $\overline{D}$ with $\overline{D}_{ii} = \sum_{i=1}^{n} |A_{ij}|$.

Signed Clustering: Signed Laplacians & Balanced Ratio Cuts

- Kunegis et al. (2010) solved a signed version of the 2-way ratio-cut problem via the Signed (combinatorial) graph Laplacian $\overline{L} = \overline{D} - A$.
  - the random-walk normalized Laplacian $\overline{L}_{\text{rw}} = I - \overline{D}^{-1} A$
  - the symmetric graph Laplacian $\overline{L}_{\text{sym}} = I - \overline{D}^{-1/2} A \overline{D}^{-1/2}$ (skewed deg dist)
- Top (eigenvalues, eigenvectors) of the Signed Laplacians contain information on the topological structure of the network. No guarantees.

Signed Clustering: Signed Laplacians & Balanced Ratio Cuts

- Kunegis et al. (2010) solved a signed version of the 2-way ratio-cut problem via the Signed (combinatorial) graph Laplacian $\overline{L} = \overline{D} - A$.
  - the random-walk normalized Laplacian $\overline{L}_{rw} = I - \overline{D}^{-1} A$
  - the symmetric graph Laplacian $\overline{L}_{sym} = I - \overline{D}^{-1/2} A \overline{D}^{-1/2}$ (skewed deg dist)
- Top (eigenvalues, eigenvectors) of the Signed Laplacians contain information on the topological structure of the network. No guarantees.

- Chiang et al (2012) considered $A = A^+ - A^-$

$$D^+ - A = D^+ - (A^+ - A^-) = D^+ - A^+ + A^- = L^+ + A^- \tag{6}$$

Signed Clustering: Signed Laplacians & Balanced Ratio Cuts

- Kunegis et al. (2010) solved a signed version of the 2-way ratio-cut problem via the Signed (combinatorial) graph Laplacian $\overline{L} = \overline{D} - A$.
  - the random-walk normalized Laplacian $\overline{L}_{rw} = I - \overline{D}^{-1} A$
  - the symmetric graph Laplacian $\overline{L}_{sym} = I - \overline{D}^{-1/2} A \overline{D}^{-1/2}$ (skewed deg dist)
- Top (eigenvalues, eigenvectors) of the Signed Laplacians contain information on the topological structure of the network. No guarantees.

- Chiang et al (2012) considered $A = A^+ - A^-$

$$D^+ - A = D^+ - (A^+ - A^-) = D^+ - A^+ + A^- = L^+ + A^- \tag{6}$$

- What does $x^T L^+ x$ count?
  - $x^T L^+ x$ measure total weight of positive edges across clusters (unhappy)
- What about $x^T A^- x$?
  - $x^T A^- x$ measure total weight of negative edges within clusters (unhappy)

Signed Clustering: Signed Laplacians & Balanced Ratio Cuts

- Kunegis et al. (2010) solved a signed version of the 2-way ratio-cut problem via the Signed (combinatorial) graph Laplacian $\overline{L} = \overline{D} - A$.
  - the random-walk normalized Laplacian $\overline{L}_{\text{rw}} = I - \overline{D}^{-1} A$
  - the symmetric graph Laplacian $\overline{L}_{\text{sym}} = I - \overline{D}^{-1/2} A \overline{D}^{-1/2}$ (skewed deg dist)
- Top (eigenvalues, eigenvectors) of the Signed Laplacians contain information on the topological structure of the network. No guarantees.

- Chiang et al (2012) considered $A = A^+ - A^-$

$$D^+ - A = D^+ - (A^+ - A^-) = D^+ - A^+ + A^- = L^+ + A^- \tag{6}$$

- What does $x^T L^+ x$ count?
  - $x^T L^+ x$ measure total weight of positive edges across clusters (unhappy)
- What about $x^T A^- x$?
  - $x^T A^- x$ measure total weight of negative edges within clusters (unhappy)

Balanced Ratio Cut:

$$\min_{\{x_1,\ldots,x_k\} \in I} \left(\sum_{c=1}^{k} \frac{x_c^T (D^+ - A) x_c}{x_c^T x_c} \right). \tag{7}$$

- where x_c is the cluster indicator vector corresponding to cluster c.

Signed clustering: a generalized eigenproblem formulation

H: unsigned graph, adj. matrix W ($W_{ij} > 0$) for any cluster $C \subset V$

$$\mathrm{cut}_H(C, \overline{C}) := \sum_{i \in C, j \in \overline{C}} W_{ij}$$

the total weight of edges crossing from C to $\overline{C}$.

Volume(C): sum of degrees of nodes in C; $\mathrm{vol}_H(C) = \sum_{i \in C} \sum_{j=1}^{n} W_{ij}$

Consider the decomposition $G = G^+ \cup G^-$; $\quad (A = A^+ - A^-)$

Signed clustering: a generalized eigenproblem formulation

H: unsigned graph, adj. matrix W ($W_{ij} > 0$) for any cluster $C \subset V$

$$\mathrm{cut}_H(C, \overline{C}) := \sum_{i \in C, j \in \overline{C}} W_{ij}$$

the total weight of edges crossing from C to $\overline{C}$.

Volume(C): sum of degrees of nodes in C; $\mathrm{vol}_H(C) = \sum_{i \in C} \sum_{j=1}^{n} W_{ij}$

Consider the decomposition $G = G^+ \cup G^-$; $\quad (A = A^+ - A^-)$

Constrained clustering $\Longrightarrow$ minimize the two measures of "*badness*"

$$\frac{\mathrm{cut}_{G^+}(C, \overline{C})}{\mathrm{vol}_{G^+}(C)}, \tag{8}$$

Signed clustering: a generalized eigenproblem formulation

H: unsigned graph, adj. matrix W ($W_{ij} > 0$) for any cluster $C \subset V$

$$\mathrm{cut}_H(C, \overline{C}) := \sum_{i \in C, j \in \overline{C}} W_{ij}$$

the total weight of edges crossing from C to $\overline{C}$.

Volume(C): sum of degrees of nodes in C; $\mathrm{vol}_H(C) = \sum_{i \in C} \sum_{j=1}^{n} W_{ij}$

Consider the decomposition $G = G^+ \cup G^-$; $\quad (A = A^+ - A^-)$

Constrained clustering $\Longrightarrow$ minimize the two measures of "*badness*"

$$\frac{\mathrm{cut}_{G^+}(C, \overline{C})}{\mathrm{vol}_{G^+}(C)}, \tag{8}$$

$$\left(\frac{\mathrm{cut}_{G^-}(C, \overline{C})}{\mathrm{vol}_{G^-}(C)} \right)^{-1} = \frac{\mathrm{vol}_{G^-}(C)}{\mathrm{cut}_{G^-}(C, \overline{C})}. \tag{9}$$

Signed clustering: a generalized eigenproblem formulation

H: unsigned graph, adj. matrix W ($W_{ij} > 0$) for any cluster $C \subset V$

$$\mathrm{cut}_H(C, \overline{C}) := \sum_{i \in C, j \in \overline{C}} W_{ij}$$

the total weight of edges crossing from C to $\overline{C}$.

Volume(C): sum of degrees of nodes in C; $\mathrm{vol}_H(C) = \sum_{i \in C} \sum_{j=1}^{n} W_{ij}$

Consider the decomposition $G = G^+ \cup G^-$; $\quad (A = A^+ - A^-)$

Constrained clustering $\Longrightarrow$ minimize the two measures of "*badness*"

$$\frac{\mathrm{cut}_{G^+}(C, \overline{C})}{\mathrm{vol}_{G^+}(C)}, \tag{8}$$

$$\left(\frac{\mathrm{cut}_{G^-}(C, \overline{C})}{\mathrm{vol}_{G^-}(C)} \right)^{-1} = \frac{\mathrm{vol}_{G^-}(C)}{\mathrm{cut}_{G^-}(C, \overline{C})}. \tag{9}$$

Ideally, want C s.t. both (8) and (9) are small. "Merge" obj. (8)+(9)

$$\min_{C \subset V} \frac{\mathrm{cut}_{G^+}(C, \overline{C}) + \tau^- \mathrm{vol}_{G^-}(C)}{\mathrm{cut}_{G^-}(C, \overline{C}) + \tau^+ \mathrm{vol}_{G^+}(C)}, \tag{10}$$

$\tau^+, \tau^- > 0$ denote trade-off/regularization parameters.

Signed clustering: a generalized eigenproblem formulation

Natural extension to $k > 2$ disjoint clusters $C_1, \cdots, C_k$

$$\min_{C_1,\cdots,C_k} \sum_{i=1}^{k} \frac{\mathrm{cut}_{G^+}(C_i, \overline{C_i}) + \tau^- \mathrm{vol}_{G^-}(C_i)}{\mathrm{cut}_{G^-}(C_i, \overline{C_i}) + \tau^+ \mathrm{vol}_{G^+}(C_i)}. \tag{11}$$

Signed clustering: a generalized eigenproblem formulation

Natural extension to $k > 2$ disjoint clusters $C_1, \cdots, C_k$

$$\min_{C_1,\cdots,C_k} \sum_{i=1}^{k} \frac{\mathrm{cut}_{G^+}(C_i, \overline{C_i}) + \tau^- \mathrm{vol}_{G^-}(C_i)}{\mathrm{cut}_{G^-}(C_i, \overline{C_i}) + \tau^+ \mathrm{vol}_{G^+}(C_i)}. \tag{11}$$

For a subset $C_i \subset V$, the normalized indicator vector

$$(x_{C_i})_j = \begin{cases} (\mathrm{cut}_{G^-}(C_i, \overline{C_i}) + \mathrm{vol}_{G^+}(C_i))^{-1/2}; & v_j \in C_i \\ 0; & v_j \notin C_i \end{cases} \tag{12}$$

Signed clustering: a generalized eigenproblem formulation

Natural extension to $k > 2$ disjoint clusters $C_1, \cdots, C_k$

$$\min_{C_1,\cdots,C_k} \sum_{i=1}^{k} \frac{\mathrm{cut}_{G^+}(C_i, \overline{C_i}) + \tau^- \mathrm{vol}_{G^-}(C_i)}{\mathrm{cut}_{G^-}(C_i, \overline{C_i}) + \tau^+ \mathrm{vol}_{G^+}(C_i)}. \tag{11}$$

For a subset $C_i \subset V$, the normalized indicator vector

$$(x_{C_i})_j = \begin{cases} (\mathrm{cut}_{G^-}(C_i, \overline{C_i}) + \mathrm{vol}_{G^+}(C_i))^{-1/2}; & v_j \in C_i \\ 0; & v_j \notin C_i \end{cases} \tag{12}$$

renders (11) as the discrete optimization problem

$$\min_{C_1,\cdots,C_k} \sum_{i=1}^{k} \frac{x_{C_i}^T (L^+ + \tau^- D^-) x_{C_i}}{x_{C_i}^T (L^- + \tau^+ D^+) x_{C_i}}, \tag{13}$$

which is NP-hard.

- L^+ (resp. L^-) denotes the Laplacian of G^+ (resp. G^-), and
- D^+ (resp. D^-) denotes a diagonal matrix with the degrees of G^+ (resp. G^-).

Signed clustering: a generalized eigenproblem formulation

Drop the discreteness constraint & allow each $x_{C_i} \in \mathbb{R}^n$

- new set of vectors $z_1, \ldots, z_k \in \mathbb{R}^n$ orthonormal w.r.t. $L^- + \tau^+ D^+$
  - $z_i^T (L^- + \tau^+ D^+) z_i = 1$, and
  - $z_i (L^- + \tau^+ D^+) z_j = 0$, for $i \neq j$

  leads to the following modified version of (13)

$$\min_{z_i (L^- + D^+) z_j = \delta_{ij}} \sum_{i=1}^{k} \frac{z_i^T (L^+ + \tau^- D^-) z_i}{z_i^T (L^- + \tau^+ D^+) z_i}. \tag{14}$$

Signed clustering: a generalized eigenproblem formulation

Drop the discreteness constraint & allow each $x_{C_i} \in \mathbb{R}^n$

- new set of vectors $z_1, \ldots, z_k \in \mathbb{R}^n$ orthonormal w.r.t. $L^- + \tau^+ D^+$
  - $z_i^T (L^- + \tau^+ D^+) z_i = 1$, and
  - $z_i (L^- + \tau^+ D^+) z_j = 0$, for $i \neq j$

  leads to the following modified version of (13)

$$\min_{z_i (L^- + D^+) z_j = \delta_{ij}} \sum_{i=1}^{k} \frac{z_i^T (L^+ + \tau^- D^-) z_i}{z_i^T (L^- + \tau^+ D^+) z_i}. \tag{14}$$

- choice of $(L^- + \tau^+ D^+)$-orthonormality of vectors $z_1, \ldots, z_k$ is not a relaxation of (13); leads to a suitable eigenvalue problem

Signed clustering: a generalized eigenproblem formulation

Drop the discreteness constraint & allow each $x_{C_i} \in \mathbb{R}^n$

- new set of vectors $z_1, \ldots, z_k \in \mathbb{R}^n$ orthonormal w.r.t. $L^- + \tau^+ D^+$
  - $z_i^T (L^- + \tau^+ D^+) z_i = 1$, and
  - $z_i (L^- + \tau^+ D^+) z_j = 0$, for $i \neq j$

  leads to the following modified version of (13)

$$\min_{z_i (L^- + D^+) z_j = \delta_{ij}} \sum_{i=1}^{k} \frac{z_i^T (L^+ + \tau^- D^-) z_i}{z_i^T (L^- + \tau^+ D^+) z_i}. \tag{14}$$

- choice of $(L^- + \tau^+ D^+)$-orthonormality of vectors $z_1, \ldots, z_k$ is not a relaxation of (13); leads to a suitable eigenvalue problem
- assuming $L^- + \tau^+ D^+$ full rank, consider the change of variables

$$y_i = (L^- + \tau^+ D^+)^{1/2} z_i, \tag{15}$$

- changes the orthonormality constraints of (13) to $y_i^T y_j = \delta_{ij}$.

Signed clustering: a generalized eigenproblem formulation

Drop the discreteness constraint & allow each $x_{C_i} \in \mathbb{R}^n$

- new set of vectors $z_1, \ldots, z_k \in \mathbb{R}^n$ orthonormal w.r.t. $L^- + \tau^+ D^+$
  - $z_i^T (L^- + \tau^+ D^+) z_i = 1$, and
  - $z_i (L^- + \tau^+ D^+) z_j = 0$, for $i \neq j$

  leads to the following modified version of (13)

$$\min_{z_i (L^- + D^+) z_j = \delta_{ij}} \sum_{i=1}^{k} \frac{z_i^T (L^+ + \tau^- D^-) z_i}{z_i^T (L^- + \tau^+ D^+) z_i}. \tag{14}$$

- choice of $(L^- + \tau^+ D^+)$-orthonormality of vectors $z_1, \ldots, z_k$ is not a relaxation of (13); leads to a suitable eigenvalue problem
- assuming $L^- + \tau^+ D^+$ full rank, consider the change of variables

$$y_i = (L^- + \tau^+ D^+)^{1/2} z_i, \tag{15}$$

- changes the orthonormality constraints of (13) to $y_i^T y_j = \delta_{ij}$.
- denoting matrix $Y = [y_1, \ldots, y_k] \in \mathbb{R}^{n \times k}$, one can rewrite (14) as

$$\min_{Y^T Y = I} \operatorname{Tr}\left(Y^T (L^- + \tau^+ D^+)^{-1/2} (L^+ + \tau^- D^-) (L^- + \tau^+ D^+)^{-1/2} Y \right) \tag{16}$$

Signed clustering: a generalized eigenproblem formulation

Drop the discreteness constraint & allow each $x_{C_i} \in \mathbb{R}^n$

- new set of vectors $z_1, \ldots, z_k \in \mathbb{R}^n$ orthonormal w.r.t. $L^- + \tau^+ D^+$
  - $z_i^T (L^- + \tau^+ D^+) z_i = 1$, and
  - $z_i (L^- + \tau^+ D^+) z_j = 0$, for $i \neq j$

  leads to the following modified version of (13)

$$\min_{z_i (L^- + D^+) z_j = \delta_{ij}} \sum_{i=1}^{k} \frac{z_i^T (L^+ + \tau^- D^-) z_i}{z_i^T (L^- + \tau^+ D^+) z_i}. \tag{14}$$

- choice of $(L^- + \tau^+ D^+)$-orthonormality of vectors $z_1, \ldots, z_k$ is not a relaxation of (13); leads to a suitable eigenvalue problem
- assuming $L^- + \tau^+ D^+$ full rank, consider the change of variables

$$y_i = (L^- + \tau^+ D^+)^{1/2} z_i, \tag{15}$$

- changes the orthonormality constraints of (13) to $y_i^T y_j = \delta_{ij}$.
- denoting matrix $Y = [y_1, \ldots, y_k] \in \mathbb{R}^{n \times k}$, one can rewrite (14) as

$$\min_{Y^T Y = I} \operatorname{Tr}\left(Y^T (L^- + \tau^+ D^+)^{-1/2} (L^+ + \tau^- D^-) (L^- + \tau^+ D^+)^{-1/2} Y \right) \tag{16}$$

- solution: eigenvectors to the k-smallest eigenvalues of

$$T = (L^- + \tau^+ D^+)^{-1/2} (L^+ + \tau^- D^-) (L^- + \tau^+ D^+)^{-1/2}$$

[16] Recall generalized eigenvalue problem: given matrices A and B, the problem of finding a vector x that satisfies $Ax = \lambda Bx$

[16] Recall generalized eigenvalue problem: given matrices A and B, the problem of finding a vector x that satisfies $Ax = \lambda Bx$

Algorithm 2 SPONGE (Signed Positive Over Negative Generalized Eigenproblem)

INPUT: A signed weighted graph G ($G = G^+ \cup G^-$)

1. find the smallest k generalized eigenvectors of
$$(L^+ + \tau^- D^-, L^- + \tau^+ D^+)$$
for suitably chosen $\tau^+, \tau^- > 0$
2. cluster the resulting embedding of the vertices in $\mathbb{R}^k$ using k-means++

---

[16] Recall generalized eigenvalue problem: given matrices A and B, the problem of finding a vector x that satisfies $Ax = \lambda Bx$

Algorithm 3 SPONGE (Signed Positive Over Negative Generalized Eigenproblem)

INPUT: A signed weighted graph G ($G = G^+ \cup G^-$)

1. find the smallest k generalized eigenvectors of
$$(L^+ + \tau^- D^-, L^- + \tau^+ D^+)$$
for suitably chosen $\tau^+, \tau^- > 0$
2. cluster the resulting embedding of the vertices in $\mathbb{R}^k$ using k-means++

SPONGE$_{sym}$: uses the smallest k generalized eigenvectors of
$$(L^+_{sym} + \tau^- I, L^-_{sym} + \tau^+ I)$$
where
$$L^+_{sym} = (D^+)^{-1/2} L^+ (D^+)^{-1/2}$$
is the symmetric Laplacian of G^+ (similarly for L^-_{sym}).

[16] Recall generalized eigenvalue problem: given matrices A and B, the problem of finding a vector x that satisfies $Ax = \lambda Bx$

Algorithm 4 SPONGE (Signed Positive Over Negative Generalized Eigenproblem)

INPUT: A signed weighted graph G ($G = G^+ \cup G^-$)

1. find the smallest k generalized eigenvectors of
$$(L^+ + \tau^- D^-, L^- + \tau^+ D^+)$$
for suitably chosen $\tau^+, \tau^- > 0$
2. cluster the resulting embedding of the vertices in $\mathbb{R}^k$ using k-means++

SPONGE$_{sym}$: uses the smallest k generalized eigenvectors of
$$(L^+_{sym} + \tau^- I, L^-_{sym} + \tau^+ I)$$
where
$$L^+_{sym} = (D^+)^{-1/2} L^+ (D^+)^{-1/2}$$
is the symmetric Laplacian of G^+ (similarly for L^-_{sym}).

• In experiments, we use LOBPCG – a preconditioned eigensolver for solving large positive definite generalized eigenproblems.

A Signed Stochastic Block Model (SSBM)

- Generate a $G(n, p)$ and partition vertex set into k equal sized clusters.

A Signed Stochastic Block Model (SSBM)

- Generate a $G(n, p)$ and partition vertex set into k equal sized clusters.
- For each edge in same (diff.) cluster, assign $+1$ (-1). Flip the sign of each edge w.p $\eta < 1/2$.

A Signed Stochastic Block Model (SSBM)

- Generate a $G(n,p)$ and partition vertex set into k equal sized clusters.
- For each edge in same (diff.) cluster, assign $+1$ (-1). Flip the sign of each edge w.p $\eta < 1/2$.
- Adjacency matrix A: For each $i < j$, we observe

  i, j lie in **same** cluster

$$A_{ij} = \begin{cases} 1 & ; \text{ w. p } & p(1-\eta) & \text{correct} \\ -1 & ; \text{ w. p } & p\eta & \text{noisy} \\ 0 & ; \text{ w. p } & (1-p) & \text{missing} \end{cases} \tag{17}$$

A Signed Stochastic Block Model (SSBM)

- Generate a $G(n,p)$ and partition vertex set into k equal sized clusters.
- For each edge in same (diff.) cluster, assign $+1$ (-1). Flip the sign of each edge w.p $\eta < 1/2$.
- Adjacency matrix A: For each $i < j$, we observe

i, j lie in **same** cluster

$$A_{ij} = \begin{cases} 1 & ; \text{ w. p } & p(1-\eta) & \text{correct} \\ -1 & ; \text{ w. p } & p\eta & \text{noisy} \\ 0 & ; \text{ w. p } & (1-p) & \text{missing} \end{cases} \tag{17}$$

i, j lie in **different** clusters

$$A_{ij} = \begin{cases} 1 & ; \text{ w. p } & p\eta & \text{noisy} \\ -1 & ; \text{ w. p } & p(1-\eta) & \text{correct} \\ 0 & ; \text{ w. p } & (1-p) & \text{missing} \end{cases} . \tag{18}$$

A Signed Stochastic Block Model (SSBM)

- Generate a $G(n,p)$ and partition vertex set into k equal sized clusters.
- For each edge in same (diff.) cluster, assign $+1$ (-1). Flip the sign of each edge w.p $\eta < 1/2$.
- Adjacency matrix A: For each $i < j$, we observe

i, j lie in **same** cluster

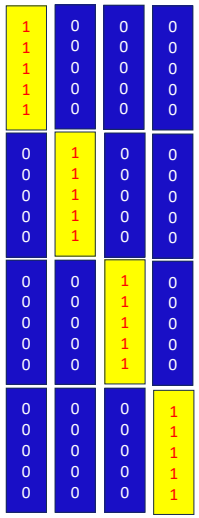
$$A_{ij} = \begin{cases} 1 & ; \text{ w. p } p(1-\eta) \text{ correct} \\ -1 & ; \text{ w. p } p\eta \text{ noisy} \\ 0 & ; \text{ w. p } (1-p) \text{ missing} \end{cases} \tag{17}$$

i, j lie in **different** clusters

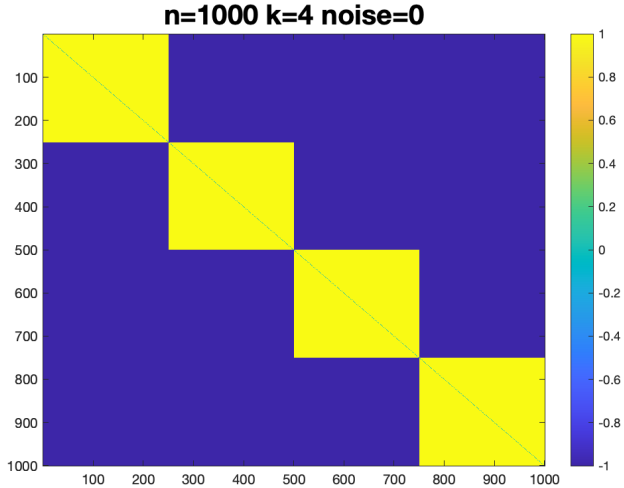
$$A_{ij} = \begin{cases} 1 & ; \text{ w. p } p\eta \text{ noisy} \\ -1 & ; \text{ w. p } p(1-\eta) \text{ correct} \\ 0 & ; \text{ w. p } (1-p) \text{ missing} \end{cases} . \tag{18}$$

- performance guarantees (misclustering rate) for SPONGE for SSBM ($k \geq 2$)

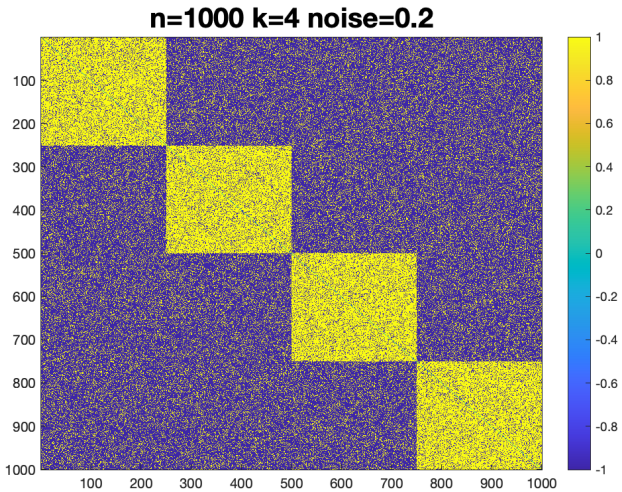
SSBM instances at varying noise levels



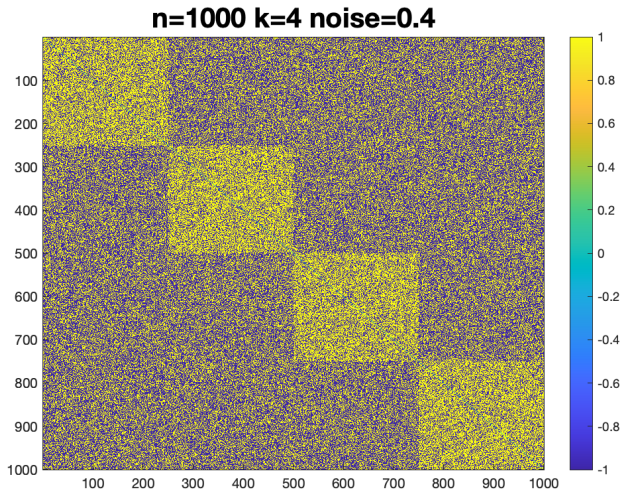
(a) Cluster membership matrix



(b) $\eta = 0$



(c) $\eta = 0.2$



(d) $\eta = 0.4$

Figure: Instances of SSBM with $n = 1000$ nodes, $k = 4$ clusters, $p = 1$ edge density.

Robustness guarantees for the SSBM

- consider the embedding given by the k smallest eigenvectors of

$$T = (L^- + \tau^+ D^+)^{-1/2}(L^+ + \tau^- D^-)(L^- + \tau^+ D^+)^{-1/2}.$$

denote

$$\overline{T} = (\mathbb{E}[L^-] + \tau^+ \mathbb{E}[D^+])^{-1/2}(\mathbb{E}[L^+] + \tau^- \mathbb{E}[D^-])(\mathbb{E}[L^-] + \tau^+ \mathbb{E}[D^+])^{-1/2}.$$

Robustness guarantees for the SSBM

- consider the embedding given by the k smallest eigenvectors of

$$T = (L^- + \tau^+ D^+)^{-1/2}(L^+ + \tau^- D^-)(L^- + \tau^+ D^+)^{-1/2}.$$

denote

$$\overline{T} = (\mathbb{E}[L^-] + \tau^+ \mathbb{E}[D^+])^{-1/2}(\mathbb{E}[L^+] + \tau^- \mathbb{E}[D^-])(\mathbb{E}[L^-] + \tau^+ \mathbb{E}[D^+])^{-1/2}.$$

- for simplicity, focus on the case $k = 2$. So the planted clusters are

$$C_1 = \left\{1, \ldots, \frac{n}{2}\right\} \text{ and } C_2 = \left\{\frac{n}{2} + 1, \ldots, n\right\}.$$

$$w = \frac{1}{\sqrt{n}}(\underbrace{1, \ldots, 1}_{n/2}, \underbrace{-1, \ldots, -1}_{n/2}) \in \mathbb{R}^n$$

- $V_2(T), V_2(\overline{T}) \in \mathbb{R}^{n \times 2}$ consist of the "smallest" two eigenvectors of $T, \overline{T}$
- $\mathcal{R}(V_2(T))$ is close to $\mathcal{R}(V_2(\overline{T}))$ w.h.p provided n, p are large enough.

Robustness guarantees for the SSBM

- consider the embedding given by the k smallest eigenvectors of

$$T = (L^- + \tau^+ D^+)^{-1/2}(L^+ + \tau^- D^-)(L^- + \tau^+ D^+)^{-1/2}.$$

denote

$$\overline{T} = (\mathbb{E}[L^-] + \tau^+ \mathbb{E}[D^+])^{-1/2}(\mathbb{E}[L^+] + \tau^- \mathbb{E}[D^-])(\mathbb{E}[L^-] + \tau^+ \mathbb{E}[D^+])^{-1/2}.$$

- for simplicity, focus on the case $k = 2$. So the planted clusters are

$$C_1 = \left\{1, \ldots, \frac{n}{2}\right\} \text{ and } C_2 = \left\{\frac{n}{2} + 1, \ldots, n\right\}.$$

$$w = \frac{1}{\sqrt{n}}(\underbrace{1, \ldots, 1}_{n/2}, \underbrace{-1, \ldots, -1}_{n/2}) \in \mathbb{R}^n$$

- $V_2(T), V_2(\overline{T}) \in \mathbb{R}^{n\times 2}$ consist of the "smallest" two eigenvectors of $T, \overline{T}$
- $\mathcal{R}(V_2(T))$ is close to $\mathcal{R}(V_2(\overline{T}))$ w.h.p provided n, p are large enough.
- for $\eta \in [0, 1/2)$, if $\tau^+, \tau^- > 0$ satisfy $\tau^- < \tau^+\left(\frac{\frac{n}{2}-1+\eta}{\frac{n}{2}-\eta}\right)$ then $\left\{\frac{1}{\sqrt{n}}\mathbf{1}, w\right\}$ are smallest two eigenvectors of $\overline{T}$

Robustness guarantees for the SSBM

- consider the embedding given by the k smallest eigenvectors of

$$T = (L^- + \tau^+ D^+)^{-1/2}(L^+ + \tau^- D^-)(L^- + \tau^+ D^+)^{-1/2}.$$

denote

$$\overline{T} = (\mathbb{E}[L^-] + \tau^+ \mathbb{E}[D^+])^{-1/2}(\mathbb{E}[L^+] + \tau^- \mathbb{E}[D^-])(\mathbb{E}[L^-] + \tau^+ \mathbb{E}[D^+])^{-1/2}.$$

- for simplicity, focus on the case $k = 2$. So the planted clusters are

$$C_1 = \left\{1, \ldots, \frac{n}{2}\right\} \text{ and } C_2 = \left\{\frac{n}{2} + 1, \ldots, n\right\}.$$

$$w = \frac{1}{\sqrt{n}}(\underbrace{1, \ldots, 1}_{n/2}, \underbrace{-1, \ldots, -1}_{n/2}) \in \mathbb{R}^n$$

- $V_2(T), V_2(\overline{T}) \in \mathbb{R}^{n \times 2}$ consist of the "smallest" two eigenvectors of $T, \overline{T}$
- $\mathcal{R}(V_2(T))$ is close to $\mathcal{R}(V_2(\overline{T}))$ w.h.p provided n, p are large enough.
- for $\eta \in [0, 1/2)$, if $\tau^+, \tau^- > 0$ satisfy $\tau^- < \tau^+\left(\frac{\frac{n}{2}-1+\eta}{\frac{n}{2}-\eta}\right)$ then $\left\{\frac{1}{\sqrt{n}}\mathbf{1}, w\right\}$ are smallest two eigenvectors of $\overline{T}$
- concentration bounds for L^-, L^+, D^-, D^+ leading to bound on $||T - \overline{T}||_2$

Robustness guarantees for the SSBM

- consider the embedding given by the k smallest eigenvectors of

$$T = (L^- + \tau^+ D^+)^{-1/2}(L^+ + \tau^- D^-)(L^- + \tau^+ D^+)^{-1/2}.$$

denote

$$\overline{T} = (\mathbb{E}[L^-] + \tau^+ \mathbb{E}[D^+])^{-1/2}(\mathbb{E}[L^+] + \tau^- \mathbb{E}[D^-])(\mathbb{E}[L^-] + \tau^+ \mathbb{E}[D^+])^{-1/2}.$$

- for simplicity, focus on the case $k = 2$. So the planted clusters are

$$C_1 = \left\{1, \ldots, \frac{n}{2}\right\} \text{ and } C_2 = \left\{\frac{n}{2} + 1, \ldots, n\right\}.$$

$$w = \frac{1}{\sqrt{n}}(\underbrace{1, \ldots, 1}_{n/2}, \underbrace{-1, \ldots, -1}_{n/2}) \in \mathbb{R}^n$$

- $V_2(T), V_2(\overline{T}) \in \mathbb{R}^{n \times 2}$ consist of the "smallest" two eigenvectors of $T, \overline{T}$
- $\mathcal{R}(V_2(T))$ is close to $\mathcal{R}(V_2(\overline{T}))$ w.h.p provided n, p are large enough.
- for $\eta \in [0, 1/2)$, if $\tau^+, \tau^- > 0$ satisfy $\tau^- < \tau^+\left(\frac{\frac{n}{2}-1+\eta}{\frac{n}{2}-\eta}\right)$ then $\left\{\frac{1}{\sqrt{n}}\mathbf{1}, w\right\}$ are smallest two eigenvectors of $\overline{T}$
- concentration bounds for L^-, L^+, D^-, D^+ leading to bound on $||T - \overline{T}||_2$
- obtain via Davis-Kahan theorem, a bound on $\left\|\sin\Theta(V_2(T), V_2(\overline{T}))\right\|_2$

Robustness guarantees for the SSBM

- consider the embedding given by the k smallest eigenvectors of

$$T = (L^- + \tau^+ D^+)^{-1/2}(L^+ + \tau^- D^-)(L^- + \tau^+ D^+)^{-1/2}.$$

denote

$$\overline{T} = (\mathbb{E}[L^-] + \tau^+ \mathbb{E}[D^+])^{-1/2}(\mathbb{E}[L^+] + \tau^- \mathbb{E}[D^-])(\mathbb{E}[L^-] + \tau^+ \mathbb{E}[D^+])^{-1/2}.$$

- for simplicity, focus on the case $k = 2$. So the planted clusters are

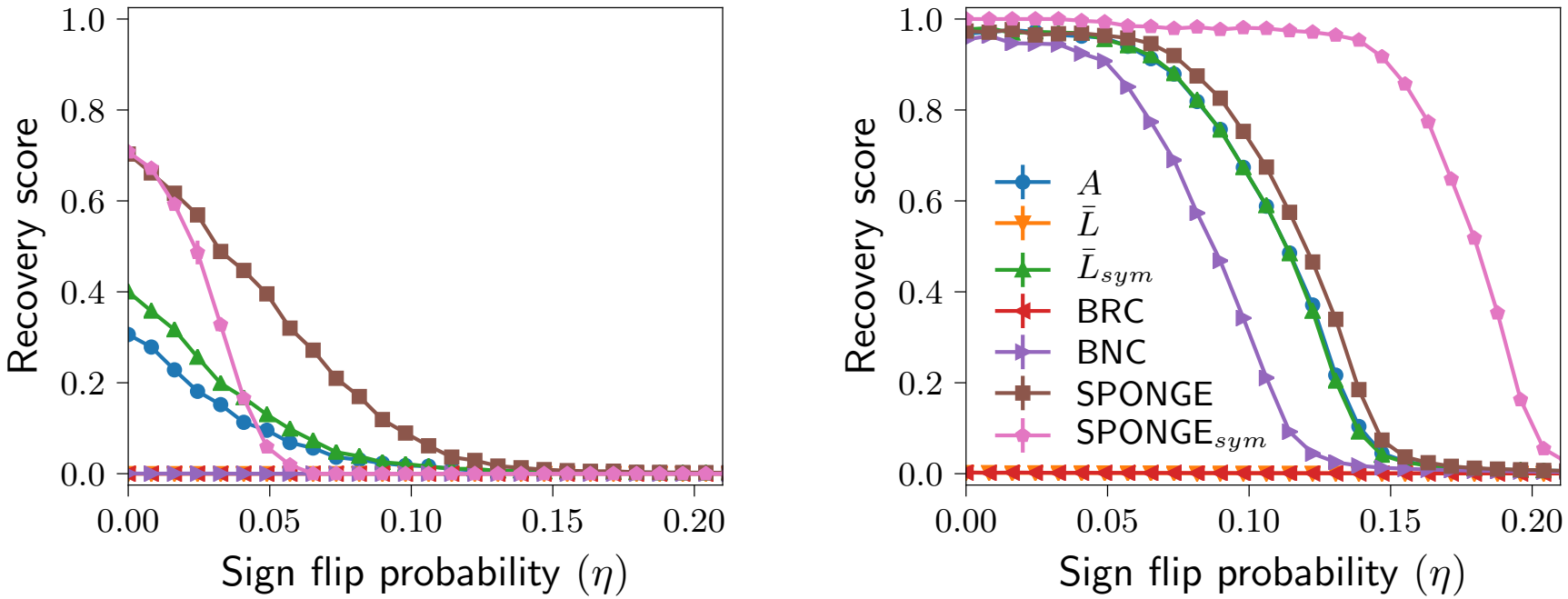
$$C_1 = \left\{1, \ldots, \frac{n}{2}\right\} \text{ and } C_2 = \left\{\frac{n}{2} + 1, \ldots, n\right\}.$$

$$w = \frac{1}{\sqrt{n}}(\underbrace{1, \ldots, 1}_{n/2}, \underbrace{-1, \ldots, -1}_{n/2}) \in \mathbb{R}^n$$

- $V_2(T), V_2(\overline{T}) \in \mathbb{R}^{n \times 2}$ consist of the "smallest" two eigenvectors of $T, \overline{T}$
- $\mathcal{R}(V_2(T))$ is close to $\mathcal{R}(V_2(\overline{T}))$ w.h.p provided n, p are large enough.
- for $\eta \in [0, 1/2)$, if $\tau^+, \tau^- > 0$ satisfy $\tau^- < \tau^+\left(\frac{\frac{n}{2}-1+\eta}{\frac{n}{2}-\eta}\right)$ then $\left\{\frac{1}{\sqrt{n}}\mathbf{1}, w\right\}$ are smallest two eigenvectors of $\overline{T}$
- concentration bounds for L^-, L^+, D^-, D^+ leading to bound on $||T - \overline{T}||_2$
- obtain via Davis-Kahan theorem, a bound on $\left\|\sin\Theta(V_2(T), V_2(\overline{T}))\right\|_2$
- provided the sparsity parameter $p = \Omega(\ln n/n)$; extended to $p = \Theta(1/n)$.

Numerical comparison of state-of-the-art methods

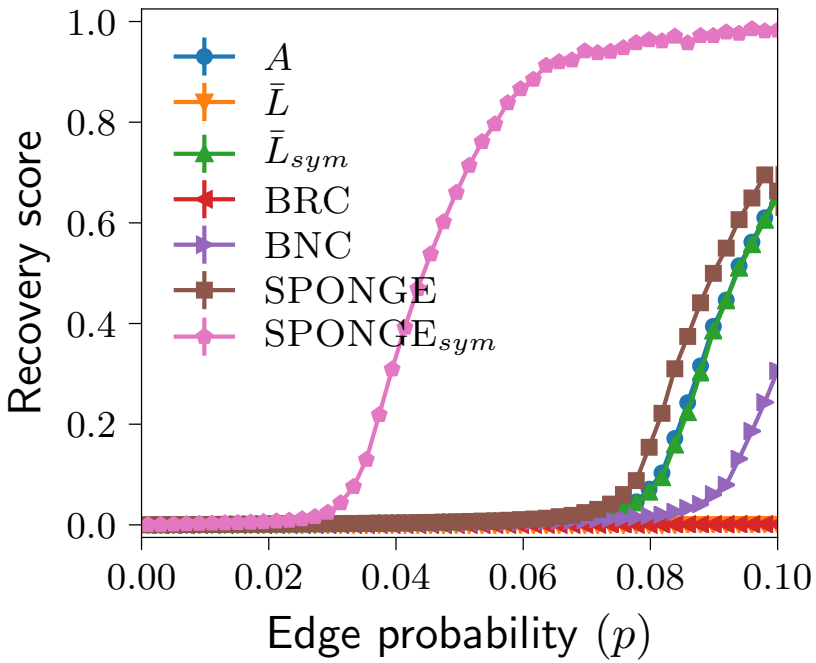
The recovery score used here is the Adjusted Rand Index (ARI) (a slightly modified version of the Rand Index).



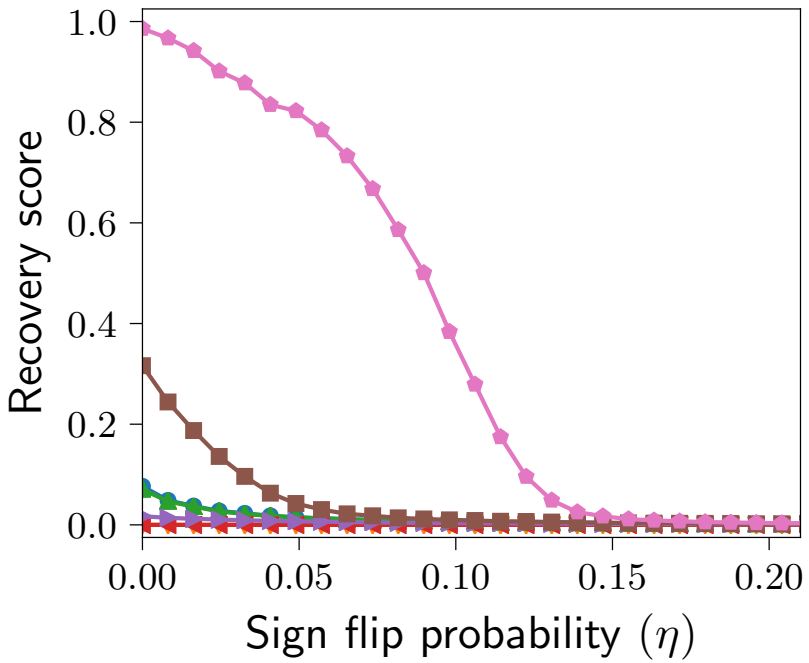
(a) $k = 5, p = 0.001$ (b) $k = 50, p = 0.1$

Figure: ARI recovery scores versus η for increasing k, with communities of equal size and $n = 10000$.

Numerical comparison of state-of-the-art methods



(a) $k = 50, \eta = 0.1$



(b) $k = 20, p = 0.01$

Figure: ARI recovery scores for $n = 10000$, as a function of:
(Left:) the edge probability p, for $\eta = 0.1$ and $k = 50$ equally-sized clusters;
(Right:) the sign flipping probability η for $k = 20, p = 0.01$.

Financial time series clustering

- Clustering of the empirical correlation matrix of 1500 time series (stocks contained in the S&P 1500 index)

Financial time series clustering

- Clustering of the empirical correlation matrix of 1500 time series (stocks contained in the S&P 1500 index)
- Compute the bottom $k = 10$ eigenvectors of $\overline{L}$, and run a standard machine learning clustering algorithm (k-means++) to recover k clusters.

Financial time series clustering

- Clustering of the empirical correlation matrix of 1500 time series (stocks contained in the S&P 1500 index)
- Compute the bottom $k = 10$ eigenvectors of $\overline{L}$, and run a standard machine learning clustering algorithm (k-means++) to recover k clusters.

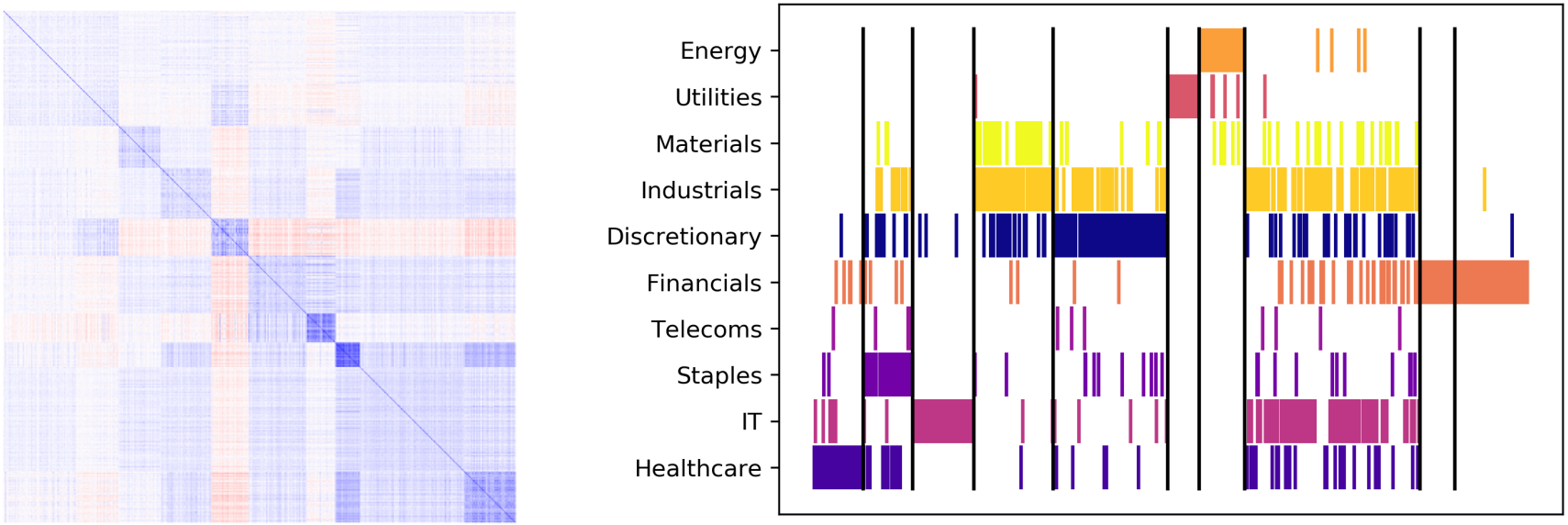
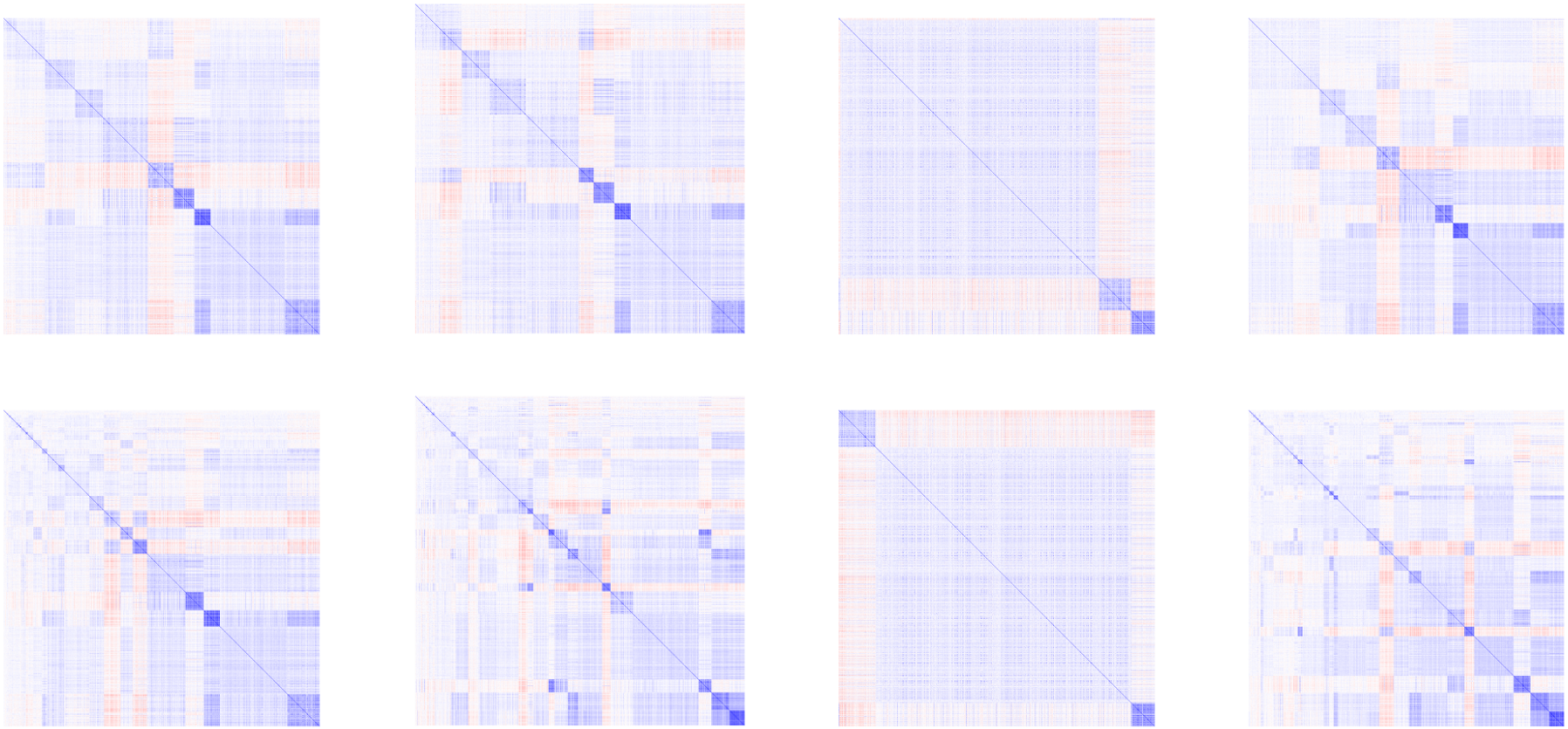


Figure: Left: the adjacency matrix A with rows/columns sorted in accordance to cluster membership. Right: Sector decomposition of the recovered clusters (based on a standard classification of the US economy into sectors).

Financial time series clustering



(e) SPONGE (f) SPONGE$_{sym}$ (g) BNC (h) $\bar{L}_{sym}$

Figure: Adjacency matrix of the S&P 1500 data, sorted by cluster membership; $k = 10$ (top) and $k = 30$ (bottom).

Cluster centric portfolios & Markowitz – Problem setup

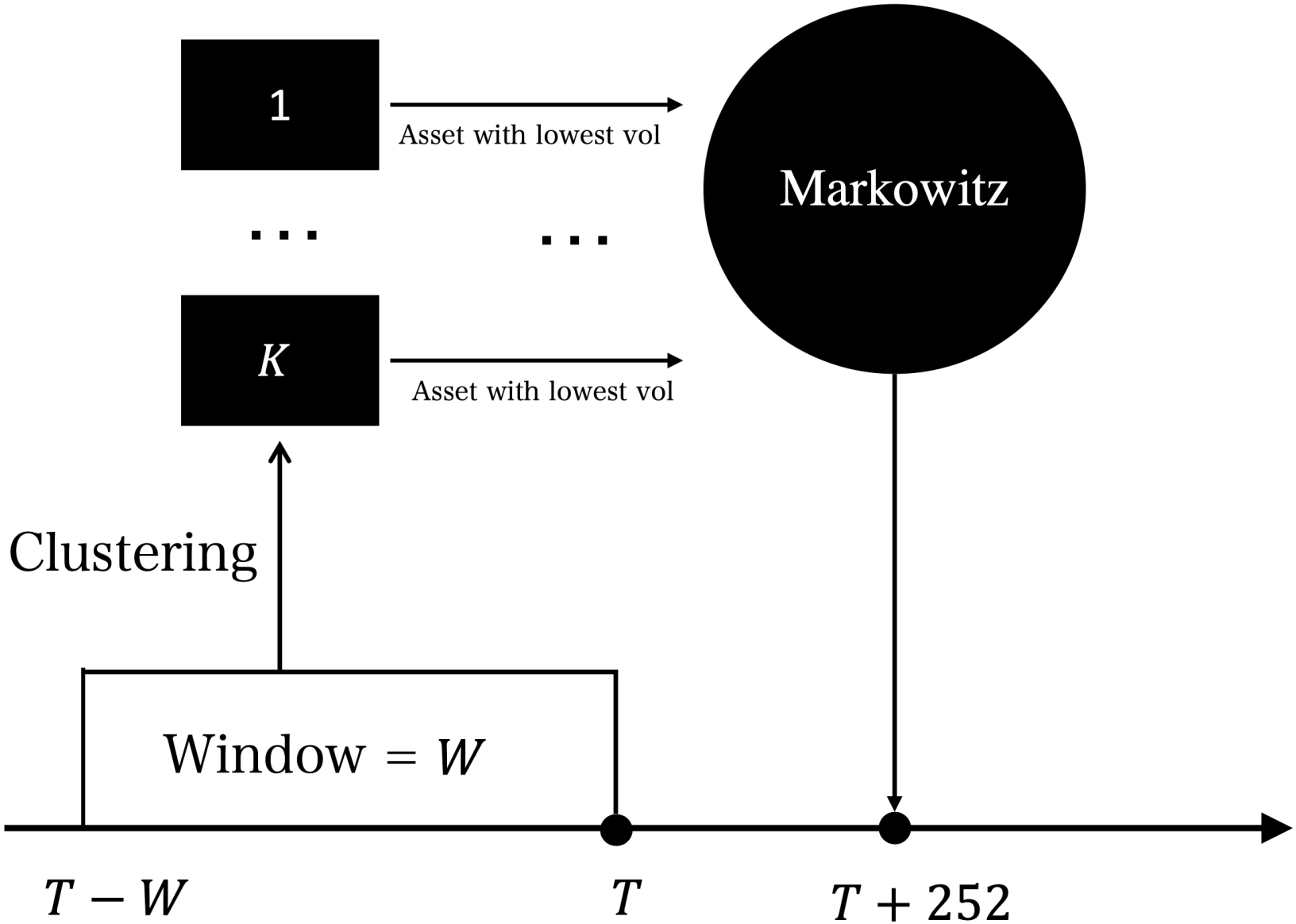


Figure: Experimental setup for the clustering and portfolio construction.

Cluster centric portfolios & Markowitz – Experimental results (I)

Table: Performance of portfolios across different clustering algorithms

Panel A: 25 clusters, 500 day lookback, 252 day holding period			
	Compound Return (%)	Return (bps/day)	Sharpe ratio
SPONGE	11.15	4.19	0.62
SPONGE$_{sym}$	8.86	3.37	0.48
Hermitian	8.76	3.33	0.52
Spectral	8.42	3.21	0.49
SPY	6.59	2.53	0.32
Panel B: 50 clusters, 500 day lookback, 252 day holding period			
	Compound Return (%)	Return (bps/day)	Sharpe ratio
SPONGE	10.62	4.00	0.58
SPONGE$_{sym}$	10.61	4.00	0.55
Hermitian	8.55	3.26	0.49
Spectral	8.06	3.08	0.44
SPY	6.59	2.53	0.32

Cluster centric portfolios & Markowitz – Experimental results (II)

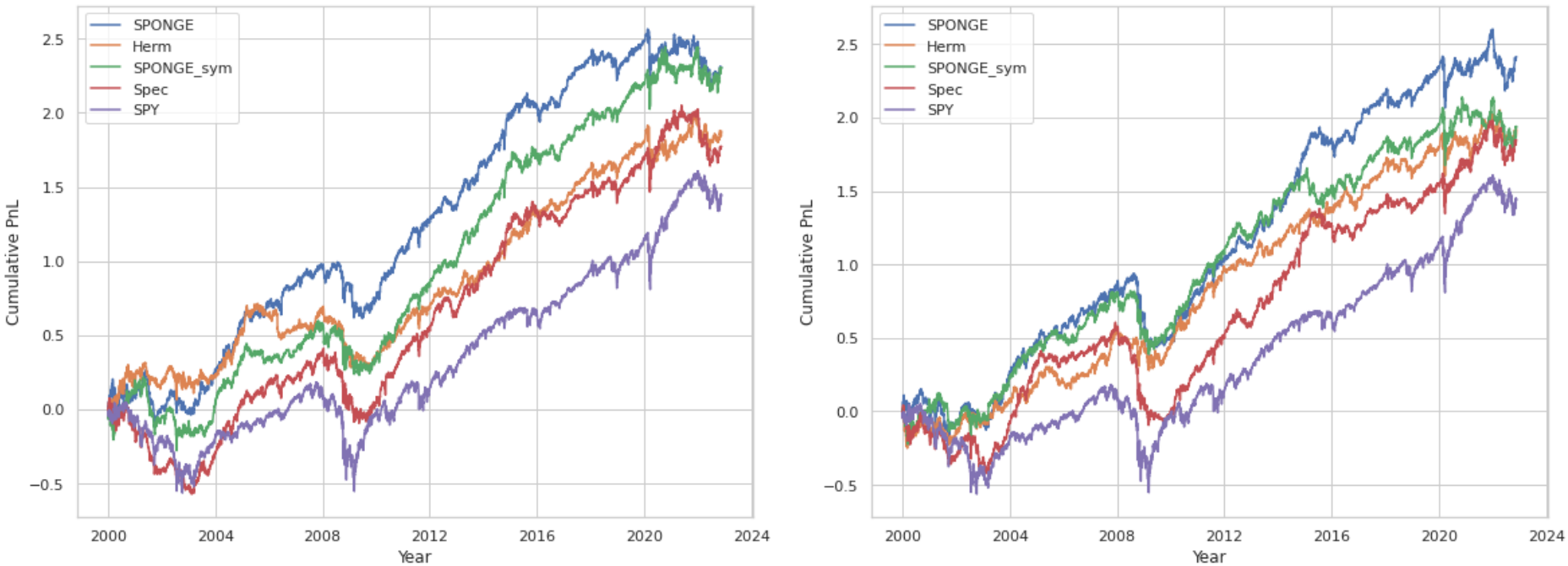


Figure: Markowitz strategy with 50 (left) and 25 (right) clusters; 252 days rebalance frequency, 500 days lookback window on the empirical correlation matrix.

Further extensions/use cases

- two-stage approach:
  - Stage I: construct an index for each cluster (with weights of assets in each cluster inversely proportional to the distance to the cluster centroid)
  - Stage II: perform classical Markowitz on the set of cluster indices.

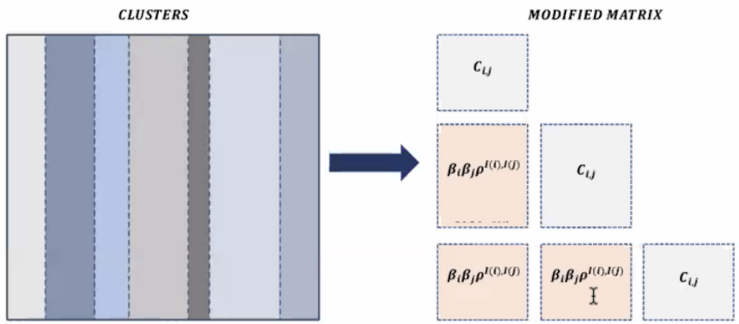
Further extensions/use cases

- two-stage approach:
  - Stage I: construct an index for each cluster (with weights of assets in each cluster inversely proportional to the distance to the cluster centroid)
  - Stage II: perform classical Markowitz on the set of cluster indices.
- statistical arbitrage settings - are residuals more likely to mean revert within clusters?

Further extensions/use cases

- two-stage approach:
  - Stage I: construct an index for each cluster (with weights of assets in each cluster inversely proportional to the distance to the cluster centroid)
  - Stage II: perform classical Markowitz on the set of cluster indices.
- statistical arbitrage settings - are residuals more likely to mean revert within clusters?
- hierarchical PCA (HPCA) (Avellaneda & Serur 2020): estimate the $n \times n$ correlation matrix based on the beta to the sector ETF;

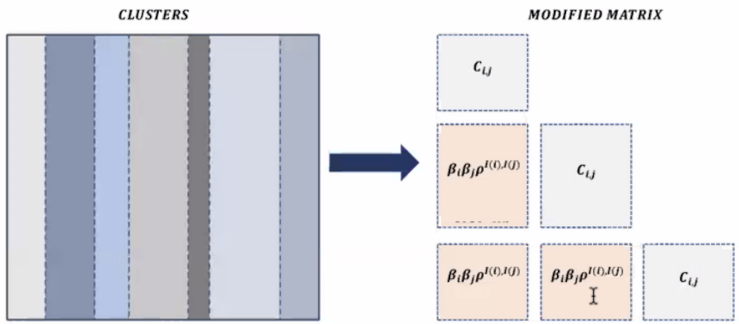
$$\widehat{C}_{i,j} = \begin{cases} C_{i,j} & \mathbb{I}(i) = \mathbb{I}(j) \\ \beta_i \beta_j \hat{\rho}^{\mathrm{I}(i),\mathbb{I}(j)} & \text{otherwise} \end{cases}$$



Further extensions/use cases

- two-stage approach:
  - Stage I: construct an index for each cluster (with weights of assets in each cluster inversely proportional to the distance to the cluster centroid)
  - Stage II: perform classical Markowitz on the set of cluster indices.
- statistical arbitrage settings - are residuals more likely to mean revert within clusters?
- hierarchical PCA (HPCA) (Avellaneda & Serur 2020): estimate the $n \times n$ correlation matrix based on the beta to the sector ETF;

$$\widehat{C}_{i,j} = \begin{cases} C_{i,j} & \mathbb{I}(i) = \mathbb{I}(j) \\ \beta_i \beta_j \hat{\rho}^{\mathrm{I}(i),\mathbb{I}(j)} & \text{otherwise} \end{cases}$$

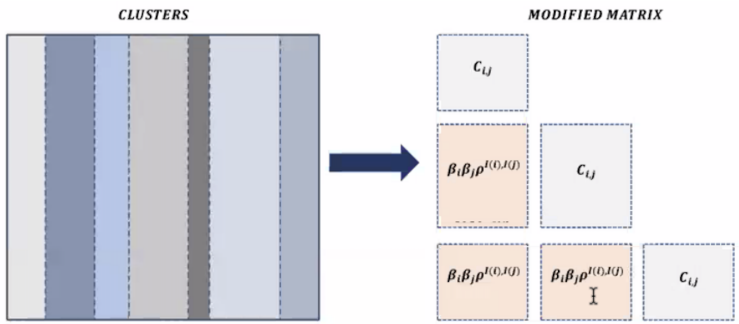


- $\mathbb{I}(i)$ is the sector if stock i; β_i is the regression coefficient of asset i on the return of the corresponding benchmark portfolio F^k (associated to sector k)
- replace the sector ETFs with data-driven (signed) clusterizations

Further extensions/use cases

- two-stage approach:
  - Stage I: construct an index for each cluster (with weights of assets in each cluster inversely proportional to the distance to the cluster centroid)
  - Stage II: perform classical Markowitz on the set of cluster indices.
- statistical arbitrage settings - are residuals more likely to mean revert within clusters?
- hierarchical PCA (HPCA) (Avellaneda & Serur 2020): estimate the $n \times n$ correlation matrix based on the beta to the sector ETF;

$$\widehat{C}_{i,j} = \begin{cases} C_{i,j} & \mathbb{I}(i) = \mathbb{I}(j) \\ \beta_i \beta_j \hat{\rho}^{\mathrm{I}(i),\mathbb{I}(j)} & \text{otherwise} \end{cases}$$

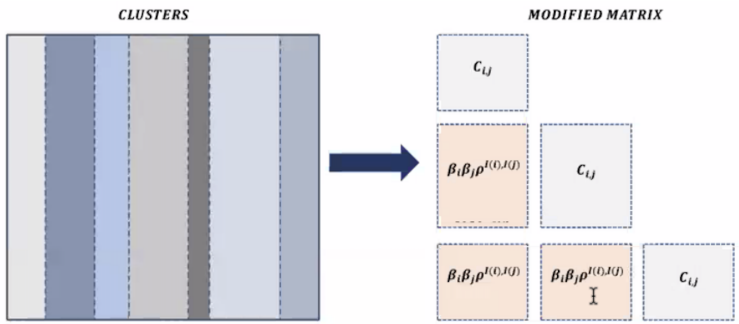


- $\mathbb{I}(i)$ is the sector if stock i; β_i is the regression coefficient of asset i on the return of the corresponding benchmark portfolio F^k (associated to sector k)
- replace the sector ETFs with data-driven (signed) clusterizations
- *"optimization approach based on statistical clustering with HPCA outperforms all the other portfolios"*

Further extensions/use cases

- two-stage approach:
  - Stage I: construct an index for each cluster (with weights of assets in each cluster inversely proportional to the distance to the cluster centroid)
  - Stage II: perform classical Markowitz on the set of cluster indices.
- statistical arbitrage settings - are residuals more likely to mean revert within clusters?
- hierarchical PCA (HPCA) (Avellaneda & Serur 2020): estimate the $n \times n$ correlation matrix based on the beta to the sector ETF;

$$\widehat{C}_{i,j} = \begin{cases} C_{i,j} & \mathbb{I}(i) = \mathbb{I}(j) \\ \beta_i \beta_j \hat{\rho}^{\mathrm{I}(i),\mathbb{I}(j)} & \text{otherwise} \end{cases}$$

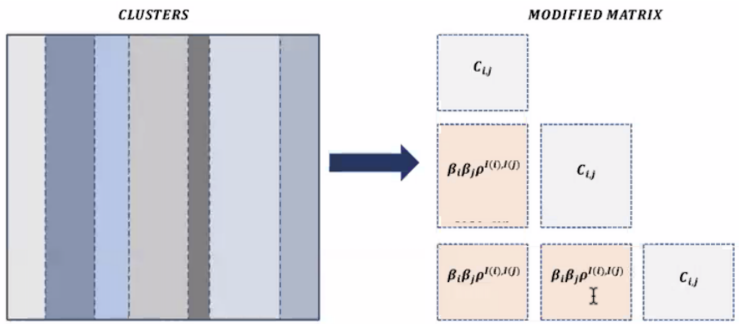


  - $\mathbb{I}(i)$ is the sector if stock i; β_i is the regression coefficient of asset i on the return of the corresponding benchmark portfolio F^k (associated to sector k)
  - replace the sector ETFs with data-driven (signed) clusterizations
  - *"optimization approach based on statistical clustering with HPCA outperforms all the other portfolios"*
- can clustering inform the factor construction procedure? (MSCI Barra/Axioma)

Further extensions/use cases

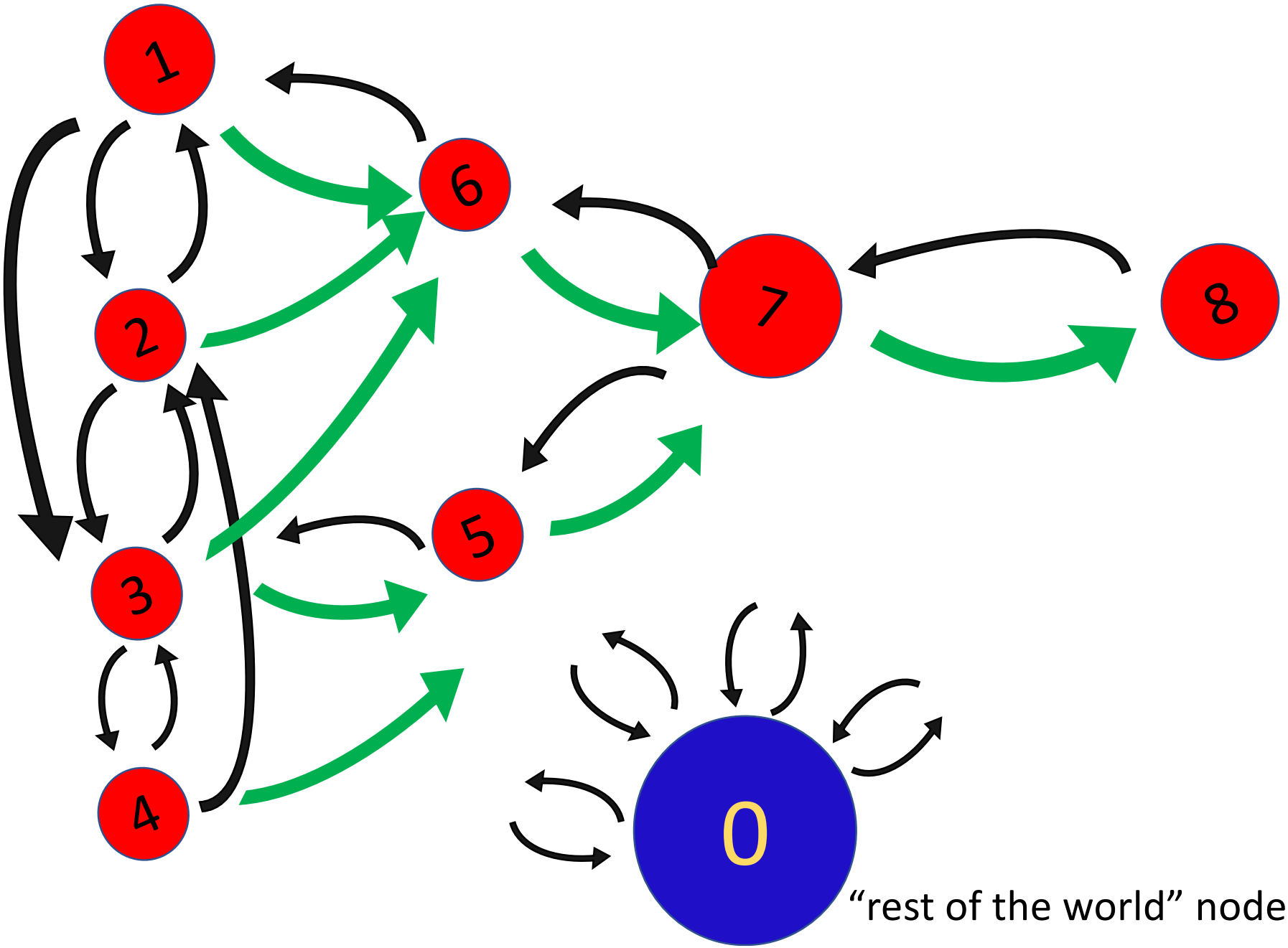
- two-stage approach:
  - Stage I: construct an index for each cluster (with weights of assets in each cluster inversely proportional to the distance to the cluster centroid)
  - Stage II: perform classical Markowitz on the set of cluster indices.
- statistical arbitrage settings - are residuals more likely to mean revert within clusters?
- hierarchical PCA (HPCA) (Avellaneda & Serur 2020): estimate the $n \times n$ correlation matrix based on the beta to the sector ETF;

$$\widehat{C}_{i,j} = \begin{cases} C_{i,j} & \mathbb{I}(i) = \mathbb{I}(j) \\ \beta_i \beta_j \hat{\rho}^{\mathrm{I}(i),\mathbb{I}(j)} & \text{otherwise} \end{cases}$$

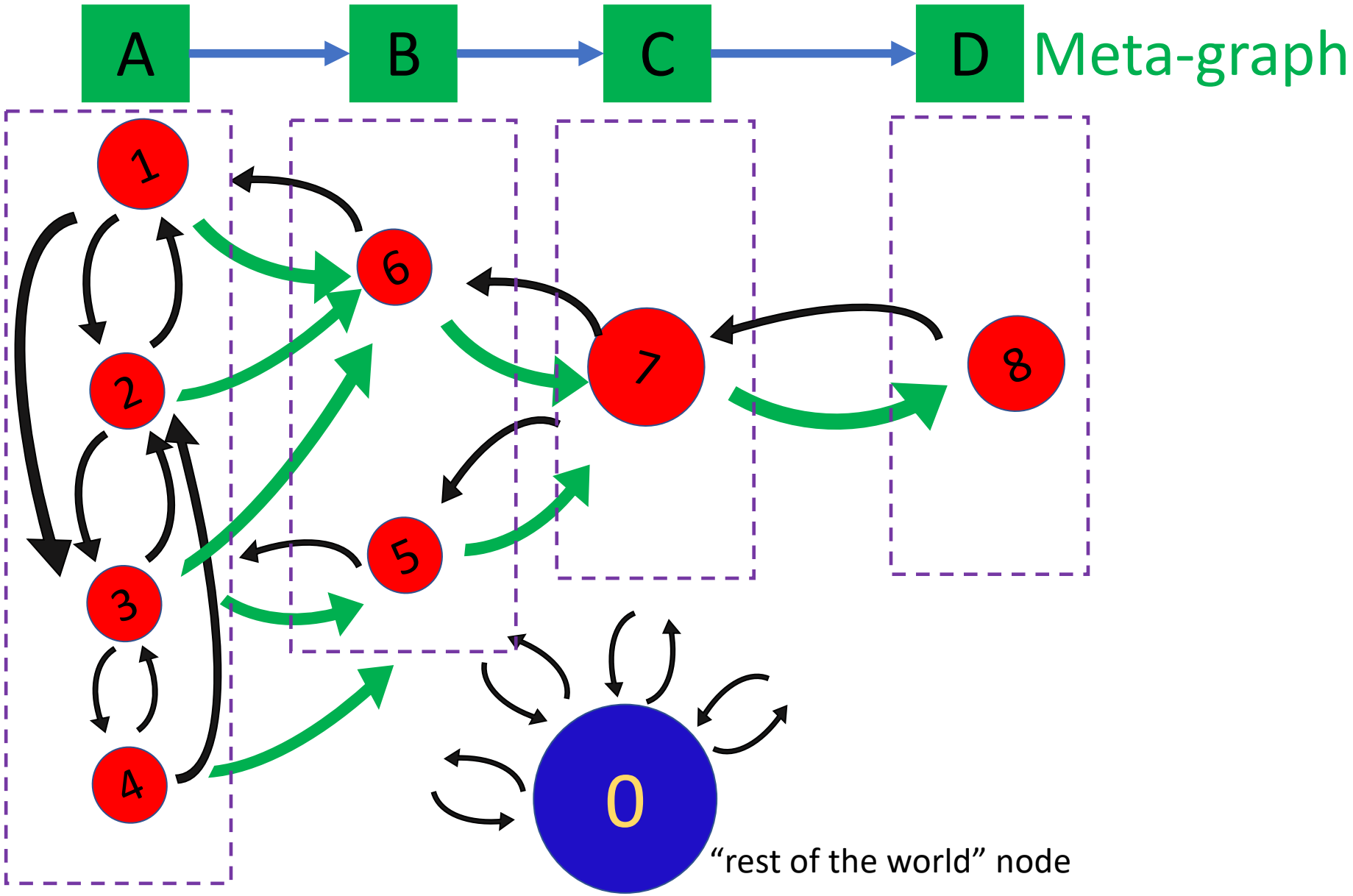


  - $\mathbb{I}(i)$ is the sector if stock i; β_i is the regression coefficient of asset i on the return of the corresponding benchmark portfolio F^k (associated to sector k)
  - replace the sector ETFs with data-driven (signed) clusterizations
  - *"optimization approach based on statistical clustering with HPCA outperforms all the other portfolios"*
- can clustering inform the factor construction procedure? (MSCI Barra/Axioma)
- covariance estimation: (Fan, Furger, Xiu 2016) truncate off-block entries outside of GICS. Why not data driven clusters?

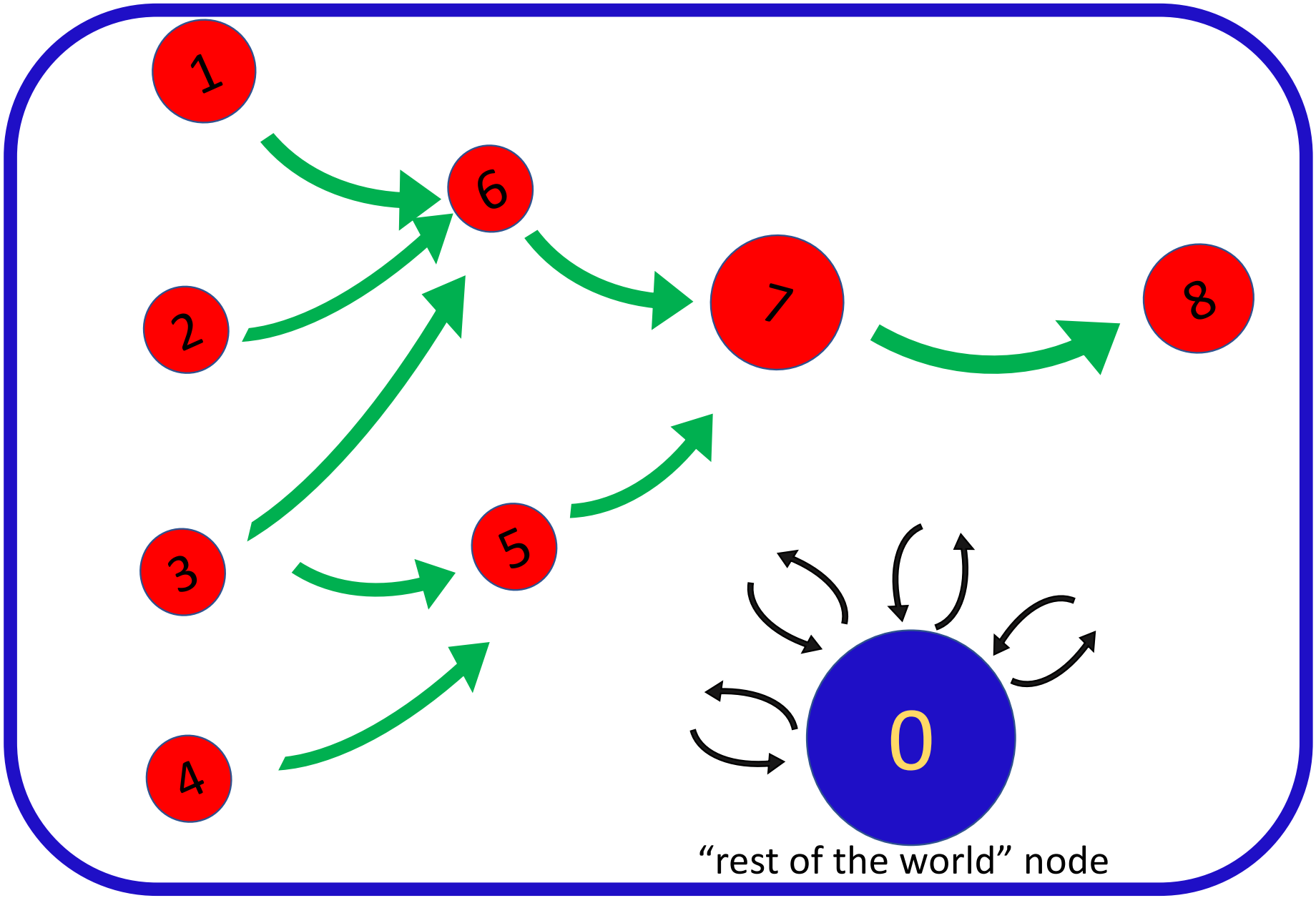
Transactions in a financial network (i)



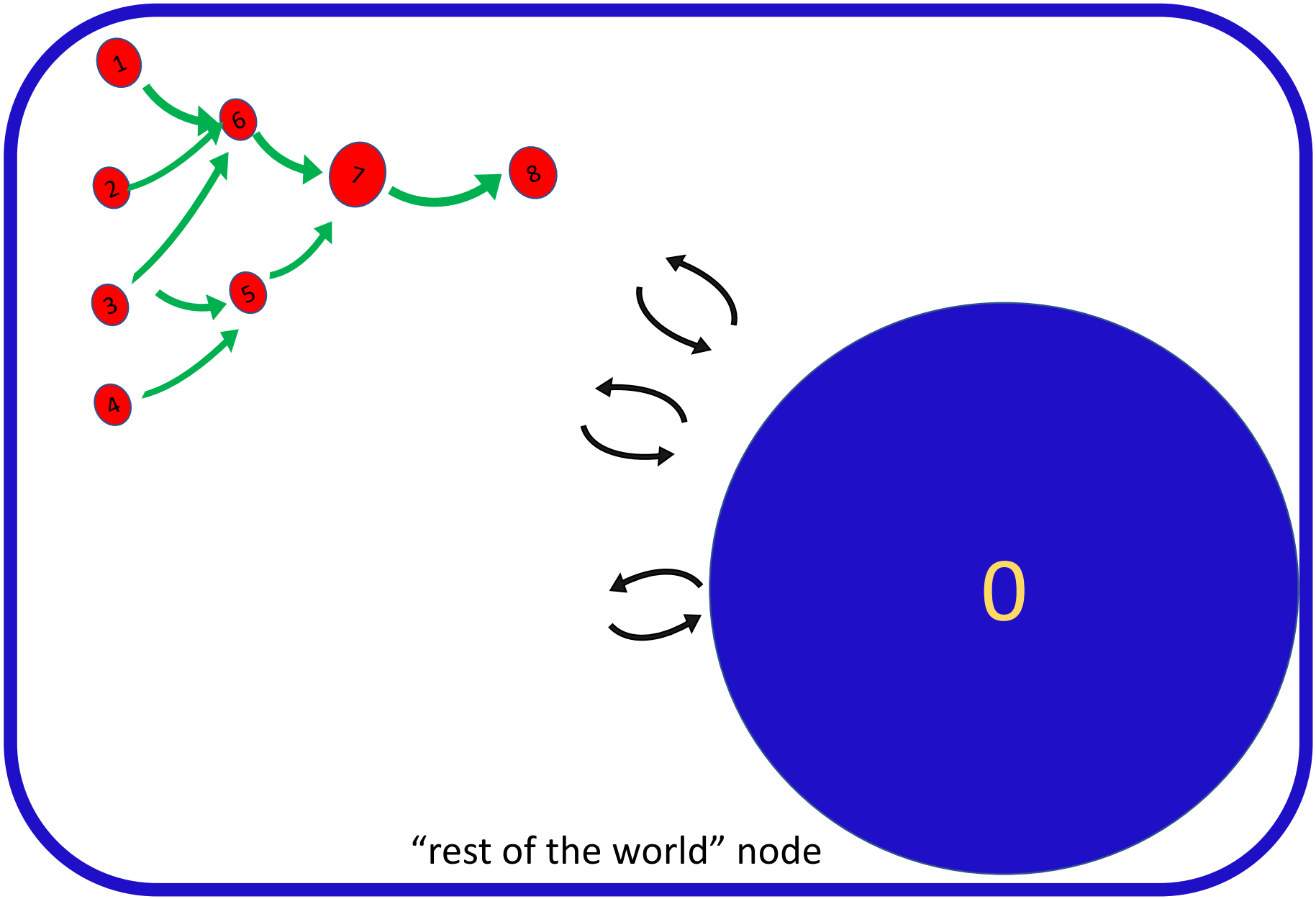
Transactions in a financial network (i)



Transactions in a financial network (ii)



Transactions in a financial network (iii)



Directed graphs - challenges

Challenges arise when handling directed graphs ($u \rightsquigarrow v$):

- usual normalized cut value (and similar clustering metrics based on edge-density) often fail to uncover many of the significant patterns

Directed graphs - challenges

Challenges arise when handling directed graphs ($u \rightsquigarrow v$):

- usual normalized cut value (and similar clustering metrics based on edge-density) often fail to uncover many of the significant patterns
- asymmetric relationships contain essential structural information about the graph

Directed graphs - challenges

Challenges arise when handling directed graphs ($u \rightsquigarrow v$):

- usual normalized cut value (and similar clustering metrics based on edge-density) often fail to uncover many of the significant patterns
- asymmetric relationships contain essential structural information about the graph
- naive symmetrization of the adjacency matrix $M \mapsto M + M^{\mathsf{T}}$

Directed graphs - challenges

Challenges arise when handling directed graphs ($u \rightsquigarrow v$):

- usual normalized cut value (and similar clustering metrics based on edge-density) often fail to uncover many of the significant patterns
- asymmetric relationships contain essential structural information about the graph
- naive symmetrization of the adjacency matrix $M \mapsto M + M^\top$

M_{ij} := number of people migrating from county i to j in the US

- migration flows btw. counties in different states will be lost in the process

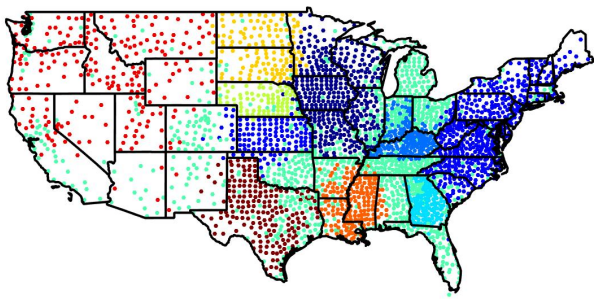
Directed graphs - challenges

Challenges arise when handling directed graphs ($u \rightsquigarrow v$):

- usual normalized cut value (and similar clustering metrics based on edge-density) often fail to uncover many of the significant patterns
- asymmetric relationships contain essential structural information about the graph
- naive symmetrization of the adjacency matrix $M \mapsto M + M^\top$

M_{ij} := number of people migrating from county i to j in the US

- migration flows btw. counties in different states will be lost in the process
- just recovers clusters that align particularly well with the political and administrative boundaries of the US states



(m) NAIVE: $M + M^\top$

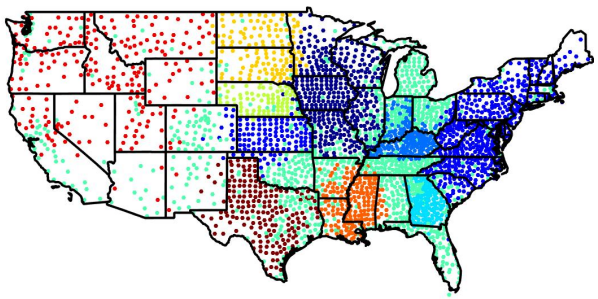
Directed graphs - challenges

Challenges arise when handling directed graphs ($u \rightsquigarrow v$):

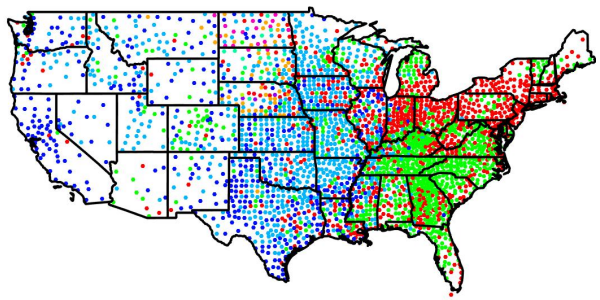
- usual normalized cut value (and similar clustering metrics based on edge-density) often fail to uncover many of the significant patterns
- asymmetric relationships contain essential structural information about the graph
- naive symmetrization of the adjacency matrix $M \mapsto M + M^\top$

M_{ij} := number of people migrating from county i to j in the US

- migration flows btw. counties in different states will be lost in the process
- just recovers clusters that align particularly well with the political and administrative boundaries of the US states



(p) NAIVE: $M + M^\top$



(q) HERMITIAN CLUSTERING

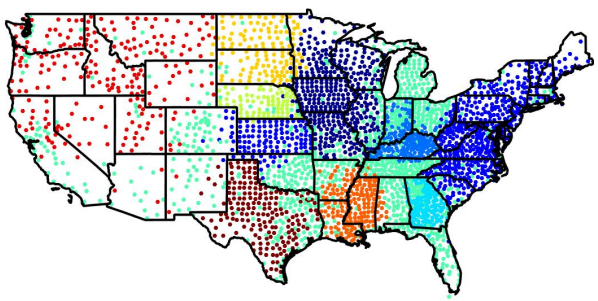
Directed graphs - challenges

Challenges arise when handling directed graphs ($u \rightsquigarrow v$):

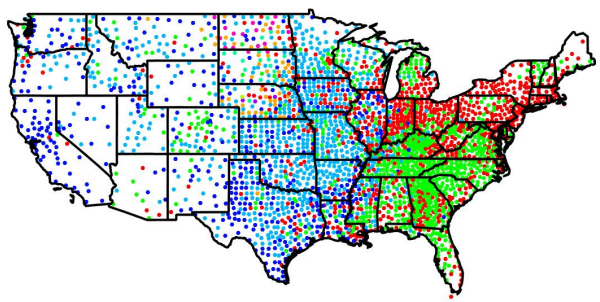
- usual normalized cut value (and similar clustering metrics based on edge-density) often fail to uncover many of the significant patterns
- asymmetric relationships contain essential structural information about the graph
- naive symmetrization of the adjacency matrix $M \mapsto M + M^\top$

M_{ij} := number of people migrating from county i to j in the US

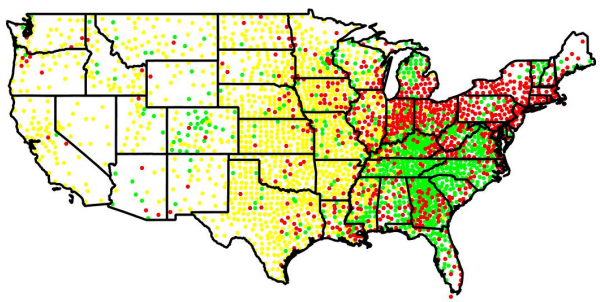
- migration flows btw. counties in different states will be lost in the process
- just recovers clusters that align particularly well with the political and administrative boundaries of the US states



(s) NAIVE: $M + M^\top$



(t) HERMITIAN CLUSTERING



(u) HERMITIAN CLUSTERING: TOP PAIR

A new algorithm for clustering directed graphs

- complex-valued Hermitian adjacency matrix $A \in \mathbb{C}^{N \times N}$ of G

$$A = (M - M^{\mathsf{T}}) \cdot i$$

- A has real eigenvalues

A new algorithm for clustering directed graphs

- complex-valued Hermitian adjacency matrix $A \in \mathbb{C}^{N \times N}$ of G

$$A = (M - M^{\mathsf{T}}) \cdot i$$

- A has real eigenvalues

- when the direction of the edges impart a cluster structure on G, this structure is approximately encoded in the eigenvectors associated with the top eigenvalues of A

A new algorithm for clustering directed graphs

- complex-valued Hermitian adjacency matrix $A \in \mathbb{C}^{N \times N}$ of G

$$A = (M - M^{\mathsf{T}}) \cdot i$$

- A has real eigenvalues

- when the direction of the edges impart a cluster structure on G, this structure is approximately encoded in the eigenvectors associated with the top eigenvalues of A

- able to capture clusters s.t. when we consider pairs of clusters, there exists a **large imbalance** in the direction of the edges from one cluster to the other

A new algorithm for clustering directed graphs

- complex-valued Hermitian adjacency matrix $A \in \mathbb{C}^{N \times N}$ of G

$$A = (M - M^{\mathsf{T}}) \cdot i$$

- A has real eigenvalues

- when the direction of the edges impart a cluster structure on G, this structure is approximately encoded in the eigenvectors associated with the top eigenvalues of A

- able to capture clusters s.t. when we consider pairs of clusters, there exists a **large imbalance** in the direction of the edges from one cluster to the other

- the Hermitian clustering algorithm uncovers the ”higher-order" structure between the clusters

A directed stochastic block model (DSBM)

Random graphs from the DSBM with parameters:

- $k \geq 2$ represents the number of clusters/communities $C_1, C_2, \ldots, C_k$

A directed stochastic block model (DSBM)

Random graphs from the DSBM with parameters:

- $k \geq 2$ represents the number of clusters/communities $C_1, C_2, \ldots, C_k$
- $\{n_j\}_{j=0}^{k-1}$ the number of vertices in each cluster

A directed stochastic block model (DSBM)

Random graphs from the DSBM with parameters:

- $k \geq 2$ represents the number of clusters/communities $C_1, C_2, \ldots, C_k$
- $\{n_j\}_{j=0}^{k-1}$ the number of vertices in each cluster
- $p \in (0, 1]$ edge probability btw. two vertices in the same cluster
- $q \in [0, 1]$ edge probability btw. two vertices in different clusters

A directed stochastic block model (DSBM)

Random graphs from the DSBM with parameters:

- $k \geq 2$ represents the number of clusters/communities $C_1, C_2, \ldots, C_k$
- $\{n_j\}_{j=0}^{k-1}$ the number of vertices in each cluster
- $p \in (0, 1]$ edge probability btw. two vertices in the same cluster
- $q \in [0, 1]$ edge probability btw. two vertices in different clusters
- want to recover clusters even when $p = q$

A directed stochastic block model (DSBM)

Random graphs from the DSBM with parameters:

- $k \geq 2$ represents the number of clusters/communities $C_1, C_2, \ldots, C_k$
- $\{n_j\}_{j=0}^{k-1}$ the number of vertices in each cluster
- $p \in (0, 1]$ edge probability btw. two vertices in the same cluster
- $q \in [0, 1]$ edge probability btw. two vertices in different clusters
- want to recover clusters even when $p = q$
- matrix $F \in [0, 1]^{k \times k}$ controls the orientations of edges between pairs of clusters

$$F_{\ell,j} + F_{j,\ell} = 1$$

A directed stochastic block model (DSBM)

Random graphs from the DSBM with parameters:

- $k \geq 2$ represents the number of clusters/communities $C_1, C_2, \ldots, C_k$
- $\{n_j\}_{j=0}^{k-1}$ the number of vertices in each cluster
- $p \in (0, 1]$ edge probability btw. two vertices in the same cluster
- $q \in [0, 1]$ edge probability btw. two vertices in different clusters
- want to recover clusters even when $p = q$
- matrix $F \in [0, 1]^{k \times k}$ controls the orientations of edges between pairs of clusters

$$F_{\ell,j} + F_{j,\ell} = 1$$

- ideally $F_{\ell,j} = 1$; $F_{j,\ell} = 0$ ($\Longrightarrow \eta = 0$ noise level)

A directed stochastic block model (DSBM)

Random graphs from the DSBM with parameters:

- $k \geq 2$ represents the number of clusters/communities $C_1, C_2, \ldots, C_k$
- $\{n_j\}_{j=0}^{k-1}$ the number of vertices in each cluster
- $p \in (0, 1]$ edge probability btw. two vertices in the same cluster
- $q \in [0, 1]$ edge probability btw. two vertices in different clusters
- want to recover clusters even when $p = q$
- matrix $F \in [0, 1]^{k \times k}$ controls the orientations of edges between pairs of clusters

$$F_{\ell,j} + F_{j,\ell} = 1$$

- ideally $F_{\ell,j} = 1$; $F_{j,\ell} = 0$ ($\Longrightarrow \eta = 0$ noise level)
- unfortunately $F_{\ell,j} = 0.90$; $F_{j,\ell} = 0.10$ ($\Longrightarrow \eta = \min(F_{\ell,j}, F_{j,\ell}) = 0.1$)

A directed stochastic block model (DSBM)

Random graphs from the DSBM with parameters:

- $k \geq 2$ represents the number of clusters/communities $C_1, C_2, \ldots, C_k$
- $\{n_j\}_{j=0}^{k-1}$ the number of vertices in each cluster
- $p \in (0, 1]$ edge probability btw. two vertices in the same cluster
- $q \in [0, 1]$ edge probability btw. two vertices in different clusters
- want to recover clusters even when $p = q$
- matrix $F \in [0, 1]^{k \times k}$ controls the orientations of edges between pairs of clusters

$$F_{\ell,j} + F_{j,\ell} = 1$$

- ideally $F_{\ell,j} = 1$; $F_{j,\ell} = 0$ ($\Longrightarrow \eta = 0$ noise level)
- unfortunately $F_{\ell,j} = 0.90$; $F_{j,\ell} = 0.10$ ($\Longrightarrow \eta = \min(F_{\ell,j}, F_{j,\ell}) = 0.1$)
- $\mathcal{G}\left(k, \{n_j\}_{j=0}^{k-1}, p, q, F\right)$

A directed stochastic block model (DSBM)

Random graphs from the DSBM with parameters:

- $k \geq 2$ represents the number of clusters/communities $C_1, C_2, \ldots, C_k$
- $\{n_j\}_{j=0}^{k-1}$ the number of vertices in each cluster
- $p \in (0, 1]$ edge probability btw. two vertices in the same cluster
- $q \in [0, 1]$ edge probability btw. two vertices in different clusters
- want to recover clusters even when $p = q$
- matrix $F \in [0, 1]^{k \times k}$ controls the orientations of edges between pairs of clusters

$$F_{\ell,j} + F_{j,\ell} = 1$$

- ideally $F_{\ell,j} = 1$; $F_{j,\ell} = 0$ ($\Longrightarrow \eta = 0$ noise level)
- unfortunately $F_{\ell,j} = 0.90$; $F_{j,\ell} = 0.10$ ($\Longrightarrow \eta = \min(F_{\ell,j}, F_{j,\ell}) = 0.1$)
- $\mathcal{G}\left(k, \{n_j\}_{j=0}^{k-1}, p, q, F\right)$
- F can be construed as the adjacency matrix of a weighted directed graph which represents the ***meta-graph*** describing the relations between the clusters

Example - directed stochastic block model (DSBM)

Matrix F - understood as the adjacency matrix of a weighted directed graph which represents the ***meta-graph*** describing relations between the clusters

$$F = \begin{pmatrix} 0.50 & 0.25 & 0.75 \\ 0.75 & 0.50 & 0.25 \\ 0.25 & 0.75 & 0.50 \end{pmatrix}$$

$$\eta = 0.25$$

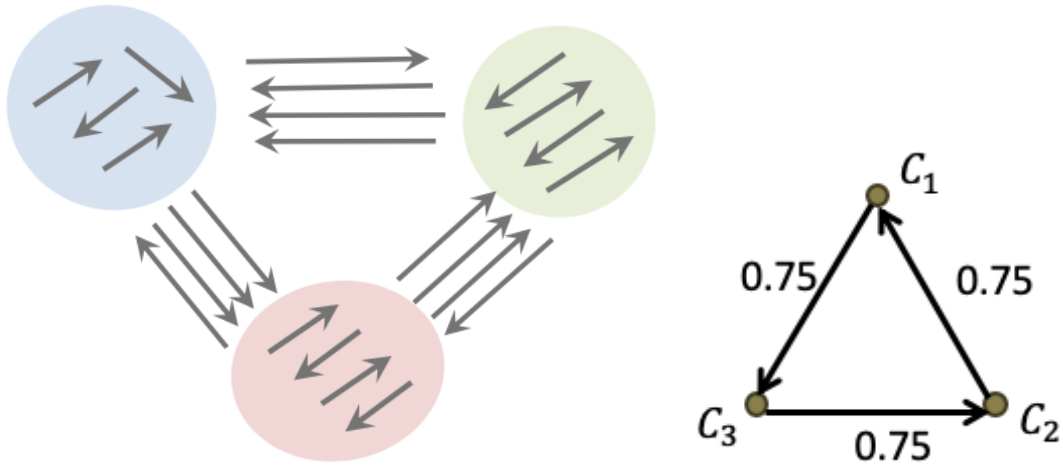


Figure: Circular flow.

Example - directed stochastic block model (DSBM)

Matrix F - understood as the adjacency matrix of a weighted directed graph which represents the ***meta-graph*** describing relations between the clusters

$$F = \begin{pmatrix} 0.50 & 0.25 & 0.75 \\ 0.75 & 0.50 & 0.25 \\ 0.25 & 0.75 & 0.50 \end{pmatrix}$$

$\eta = 0.25$

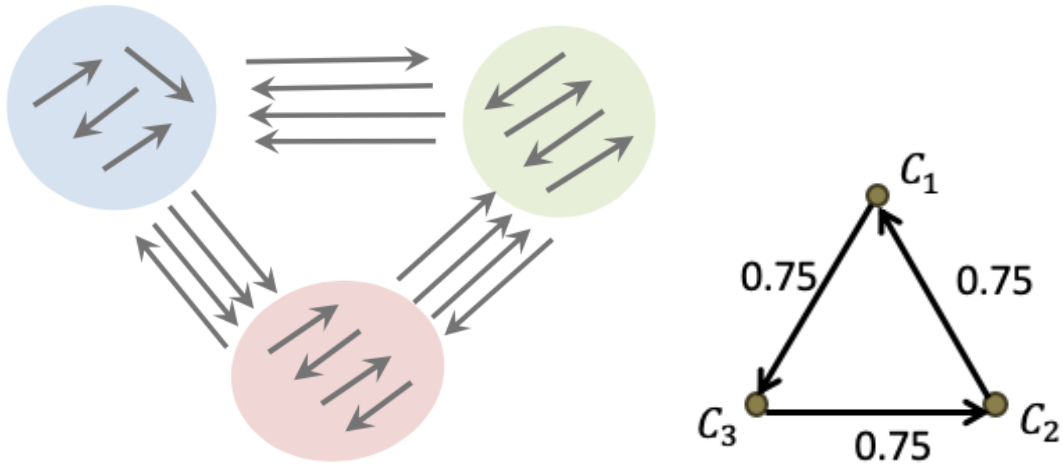


Figure: Circular flow.

- let $k = 3$, $n_1 = n_2 = n_3$, $p = q = 0.2$, and
- G consists of 3 clusters C_1, C_2 and C_3 of same size; any pair of vertices is connected by an edge with the same probability p
- directions of edges inside a cluster are chosen uniformly at random
- directions of the edges crossing different clusters are chosen non-uniformly and are defined by F.
- in expectation, all the vertices in G have the same in- and out-degrees.

Algorithm 5 Spectral clustering for a digraph

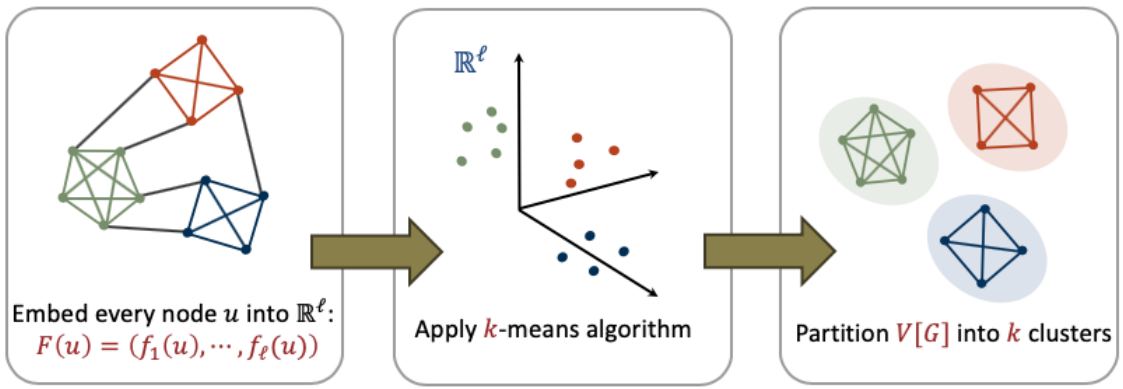
INPUT: A directed graph $G = (V, E)$, $k \geq 2$

1. Build Hermitian adjacency matrix A

$$A_{u,v} = \begin{cases} i & ; \text{ if } u \mapsto v \\ -i & ; \text{ if } v \mapsto u \\ 0 & ; \text{ if } \textit{otherwise} \end{cases} . \tag{19}$$

2. Consider the normalized Hermitian Laplacian matrix
3. Compute its top eigenvalues/eigenvectors pairs $\{(\lambda_1, f_1), \ldots, (\lambda_\ell, f_\ell)\}$
4. Apply a k-means algorithm to the resulting eigen-embedding

For general directed graphs with initial adj. matrix M set $A = (M - M^T) \cdot i$



Algorithm 6 Spectral clustering for a digraph

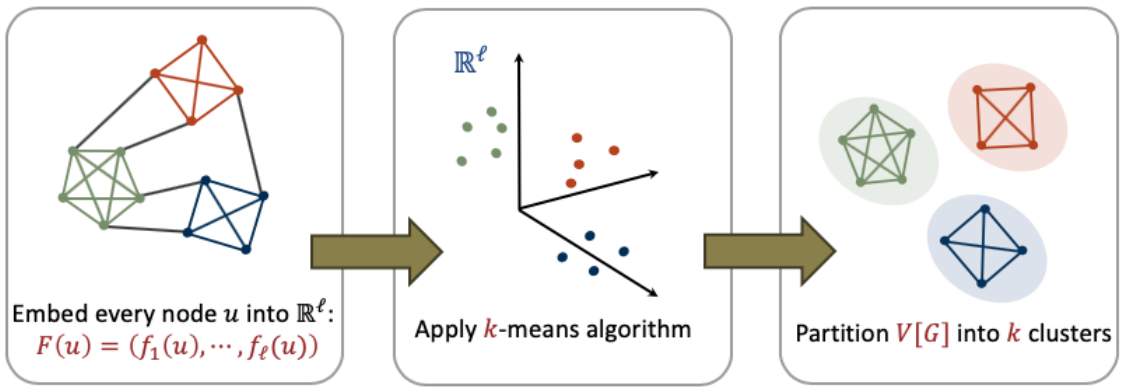
INPUT: A directed graph $G = (V, E)$, $k \geq 2$

1. Build Hermitian adjacency matrix A

$$A_{u,v} = \begin{cases} i & ; \text{ if } u \mapsto v \\ -i & ; \text{ if } v \mapsto u \\ 0 & ; \text{ if } \textit{otherwise} \end{cases} . \tag{19}$$

2. Consider the normalized Hermitian Laplacian matrix
3. Compute its top eigenvalues/eigenvectors pairs $\{(\lambda_1, f_1), \ldots, (\lambda_\ell, f_\ell)\}$
4. Apply a k-means algorithm to the resulting eigen-embedding

For general directed graphs with initial adj. matrix M set $A = (M - M^T) \cdot i$



Thm. Let $G \sim \mathcal{G}(k, n, p = q, F)$. If $\tilde{\rho} \geq C\,(k/\theta)\sqrt{(1/pn)\,\log n}$ then whp the number of misclassified vertices is $O\left(k^2/(\tilde{\rho}^2\,\theta^2\,p)\,\log n\right)$.

- Previous spectral methods count the number of common parents or children or both:

$$(M^\top M)_{uv} = |\{w : w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}|, \tag{20}$$

$$(MM^\top)_{uv} = |\{w : u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|, \tag{21}$$

$$(M^\top M + MM^\top)_{uv} = |\{w : w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}| + |\{w : u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|. \tag{22}$$

- Previous spectral methods count the number of common parents or children or both:

$$(M^\top M)_{uv} = |\{w\colon w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}|, \tag{20}$$

$$(MM^\top)_{uv} = |\{w\colon u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|, \tag{21}$$

$$\begin{aligned}(M^\top M + MM^\top)_{uv} = {} & |\{w\colon w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}| \\ & + |\{w\colon u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|.\end{aligned} \tag{22}$$

- $A = (M - M^\top) \cdot i$

$$\begin{aligned}A^2_{uv} = {} & |\{w\colon (w \rightsquigarrow u \text{ and } w \rightsquigarrow v) \text{ or } (u \rightsquigarrow w \text{ and } v \rightsquigarrow w)\}| \\ & - |\{w\colon (u \rightsquigarrow w \text{ and } w \rightsquigarrow v) \text{ or } (v \rightsquigarrow w \text{ and } w \rightsquigarrow u)\}|.\end{aligned}$$

- Previous spectral methods count the number of common parents or children or both:

$$(M^\top M)_{uv} = |\{w : w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}|, \tag{20}$$

$$(MM^\top)_{uv} = |\{w : u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|, \tag{21}$$

$$\begin{aligned}(M^\top M + MM^\top)_{uv} = {} & |\{w : w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}| \\ & + |\{w : u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|.\end{aligned} \tag{22}$$

- $A = (M - M^\top) \cdot i$

$$\begin{aligned}A^2_{uv} = {} & |\{w : (w \rightsquigarrow u \text{ and } w \rightsquigarrow v) \text{ or } (u \rightsquigarrow w \text{ and } v \rightsquigarrow w)\}| \\ & - |\{w : (u \rightsquigarrow w \text{ and } w \rightsquigarrow v) \text{ or } (v \rightsquigarrow w \text{ and } w \rightsquigarrow u)\}|.\end{aligned}$$

- A implicitly assigns a positive weight between a pair of vertices who have more common parents and offspring than ”mismatched" relations with third vertices, and a negative weight otherwise

- Previous spectral methods count the number of common parents or children or both:

$$(M^\top M)_{uv} = |\{w : w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}|, \tag{20}$$

$$(MM^\top)_{uv} = |\{w : u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|, \tag{21}$$

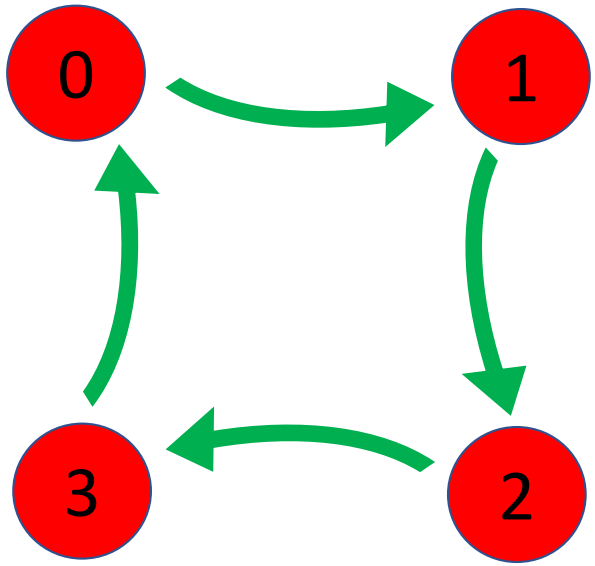
$$\begin{aligned}(M^\top M + MM^\top)_{uv} = {} & |\{w : w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}| \\ & + |\{w : u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|.\end{aligned} \tag{22}$$

- $A = (M - M^\top) \cdot i$

$$\begin{aligned}A^2_{uv} = {} & |\{w : (w \rightsquigarrow u \text{ and } w \rightsquigarrow v) \text{ or } (u \rightsquigarrow w \text{ and } v \rightsquigarrow w)\}| \\ & - |\{w : (u \rightsquigarrow w \text{ and } w \rightsquigarrow v) \text{ or } (v \rightsquigarrow w \text{ and } w \rightsquigarrow u)\}|.\end{aligned}$$

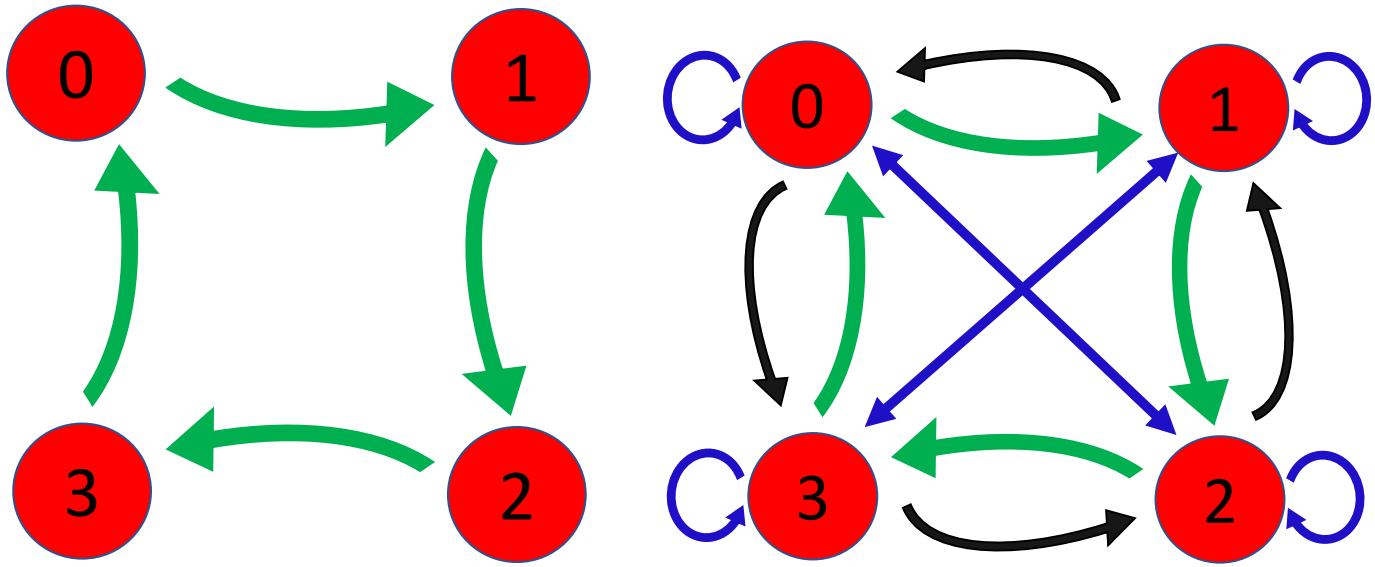
- A implicitly assigns a positive weight between a pair of vertices who have more common parents and offspring than ”mismatched" relations with third vertices, and a negative weight otherwise
- A implicitly keeps track of both common parents and offsprings without the need to perform an expensive matrix multiplication as in the case of the matrix $M^\top M + MM^\top$

Meta-graph structures on $k = 4$ nodes



(a) Clean cycle meta-graph

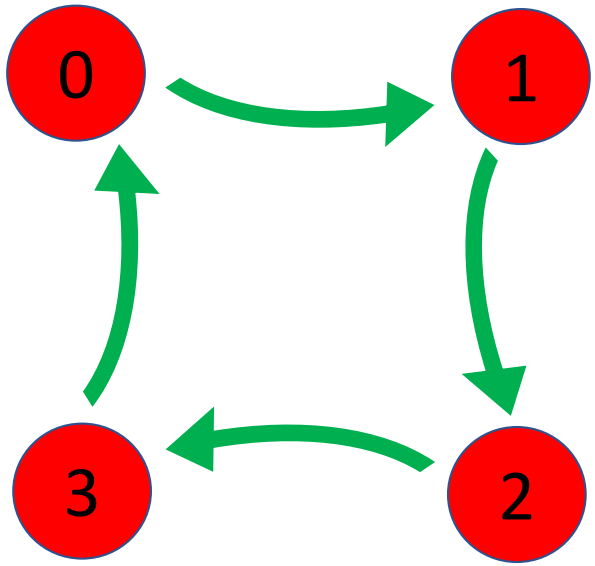
Meta-graph structures on $k = 4$ nodes



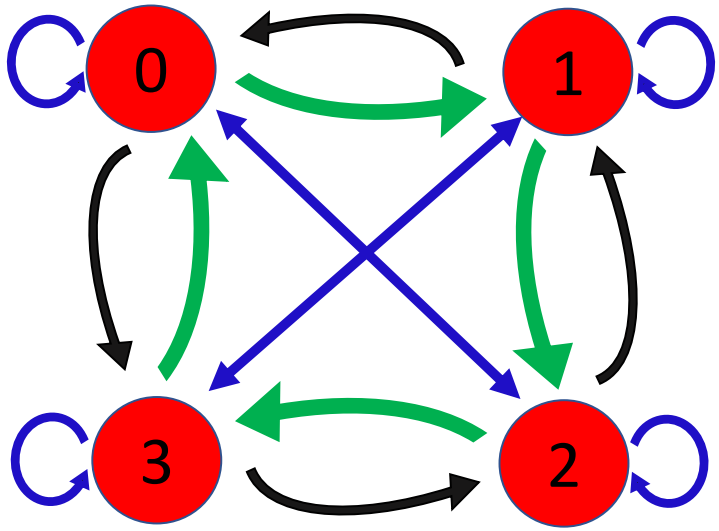
(a) Clean cycle meta-graph

(b) Noisy cycle meta-graph

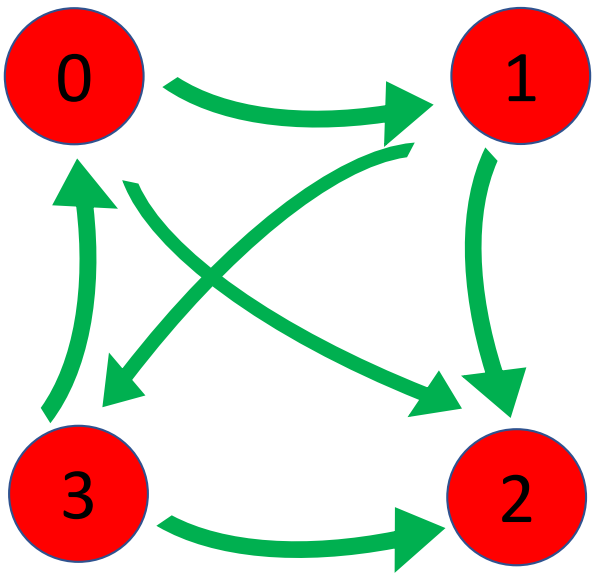
Meta-graph structures on $k = 4$ nodes



(a) Clean cycle meta-graph

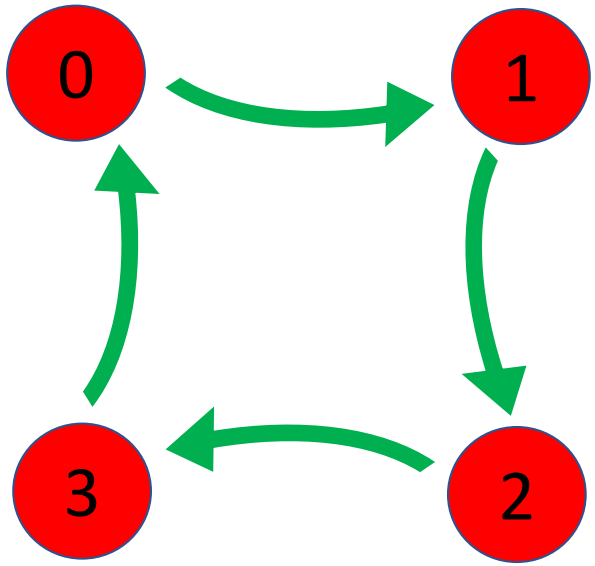


(b) Noisy cycle meta-graph

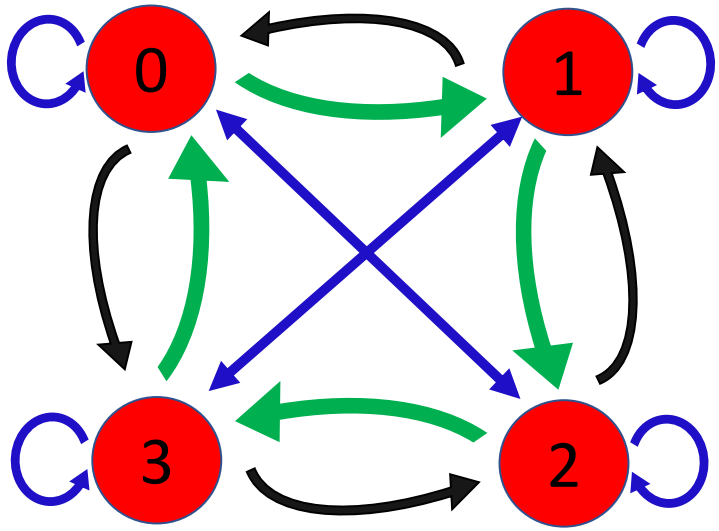


(c) Clean complete meta-graph

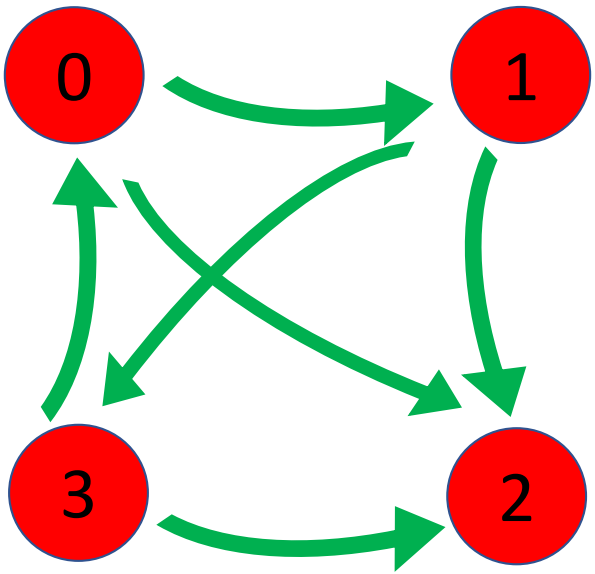
Meta-graph structures on $k = 4$ nodes



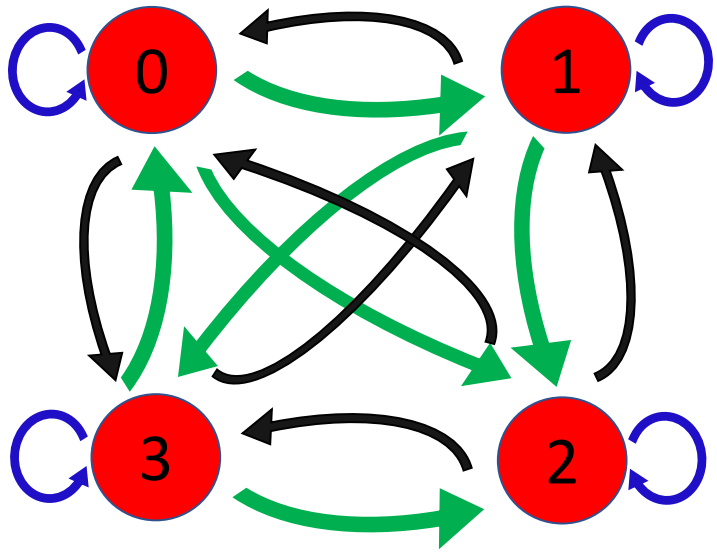
(a) Clean cycle meta-graph



(b) Noisy cycle meta-graph



(c) Clean complete meta-graph



(d) Noisy complete meta-graph

Normalization of A_G

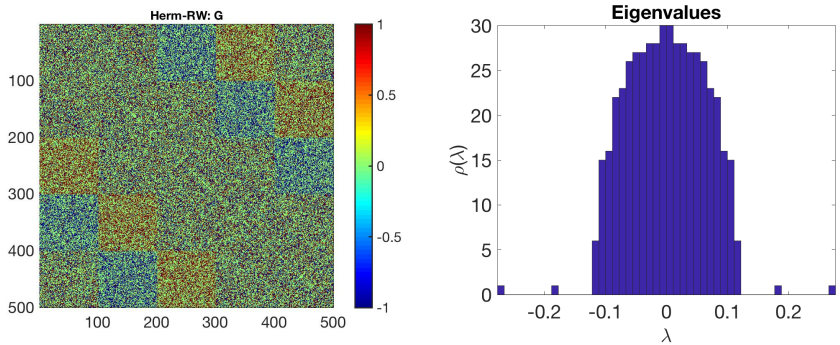
- D the diagonal matrix with $D_{jj} = \sum_{\ell=1}^{N} |A_{j\ell}|$

$$\text{HERM-RW}: \quad A_{\text{rw}} = D^{-1} A, \tag{23}$$

- similar to the Hermitian matrix

$$\text{HERM-SYM}: \quad A_{\text{sym}} = D^{-1/2} A D^{-1/2}, \tag{24}$$

- via $A_{\text{rw}} = D^{-1/2} A_{\text{sym}} D^{1/2}$
- A_{rw} also has N real eigenvalues.



(a) Cycle meta-graph: G + Spectrum

Normalization of A_G

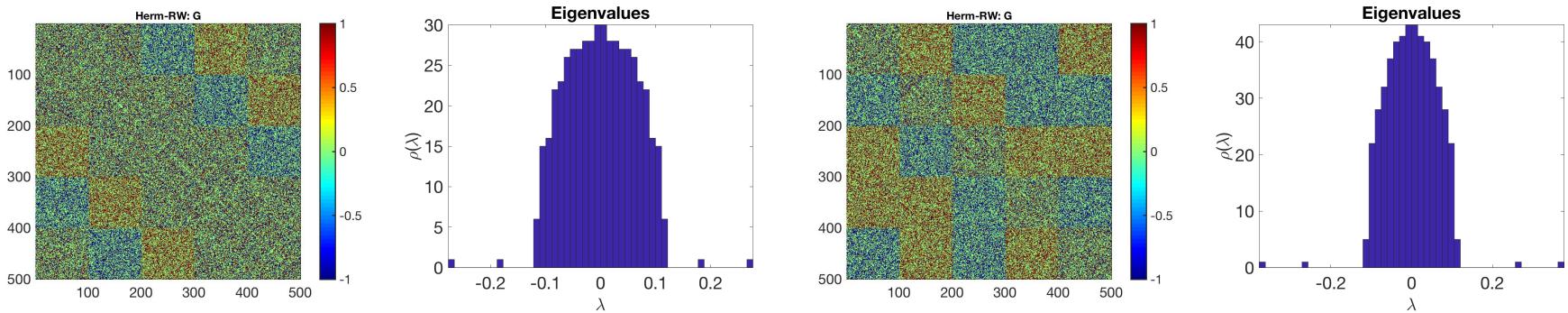
- D the diagonal matrix with $D_{jj} = \sum_{\ell=1}^{N} |A_{j\ell}|$

$$\text{HERM-RW}: \quad A_{\text{rw}} = D^{-1} A, \tag{23}$$

- similar to the Hermitian matrix

$$\text{HERM-SYM}: \quad A_{\text{sym}} = D^{-1/2} A D^{-1/2}, \tag{24}$$

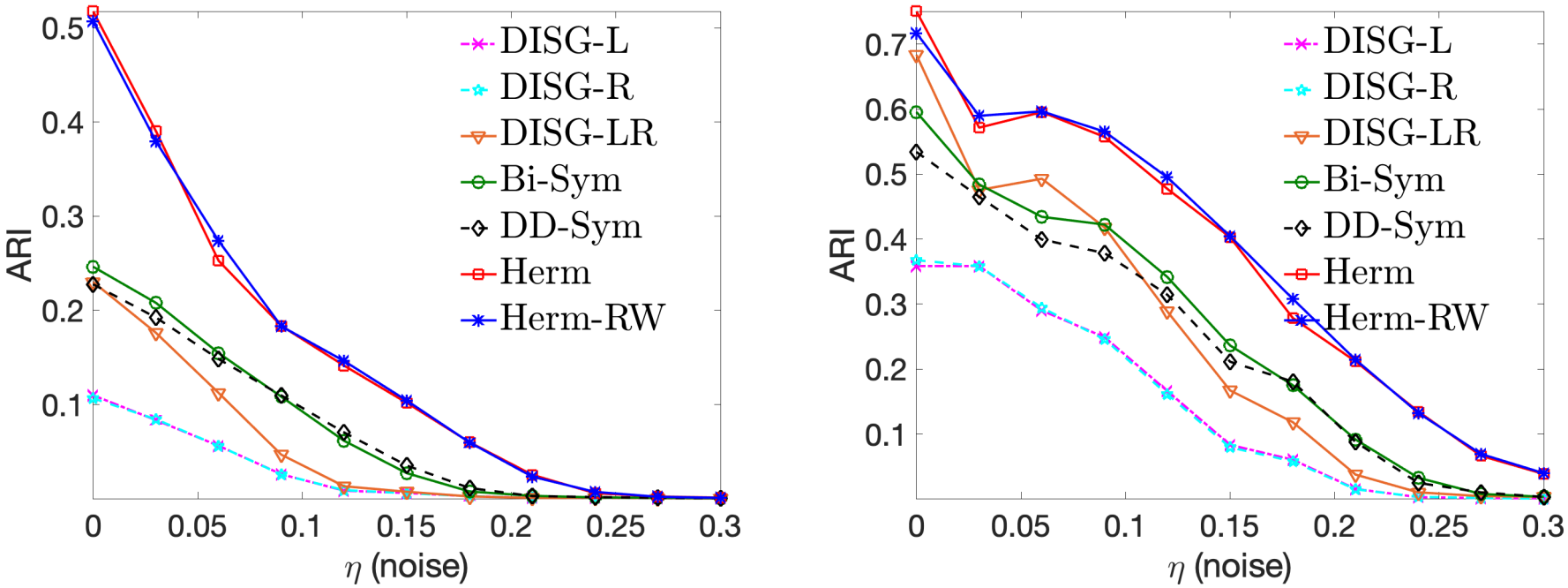
- via $A_{\text{rw}} = D^{-1/2} A_{\text{sym}} D^{1/2}$
- A_{rw} also has N real eigenvalues.



(a) Cycle meta-graph: G + Spectrum (b) Complete meta-graph: G + Spectrum

Figure: Adjacency matrices and spectra of A_{rw}.

Comparison with state-of-the-art



(a) Circular pattern (b) Complete meta-graph

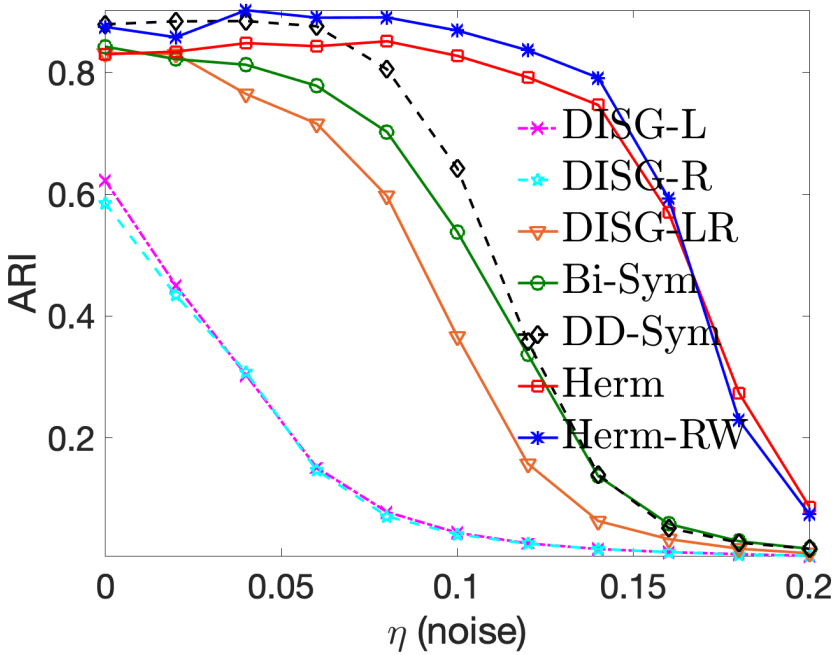
Figure: Recovery rates for the DSBM with $k = 5$, $N = 5000$, at sparsity $p = 0.5\%$. Averaged over 10 runs.

- **DISG**: K. Rohe, T. Qin, and B. Yu. *Co-clustering directed graphs to discover asymmetries and directional communities*. Proceedings of the National Academy of Sciences (2016); regularized graph Laplacian

$$L_{ij} = \frac{A_{ij}}{O^{\tau}_{ii} P^{\tau}_{jj}} = \left[(O^{\tau})^{-1/2} A (P^{\tau})^{-1/2} \right]_{ij}$$

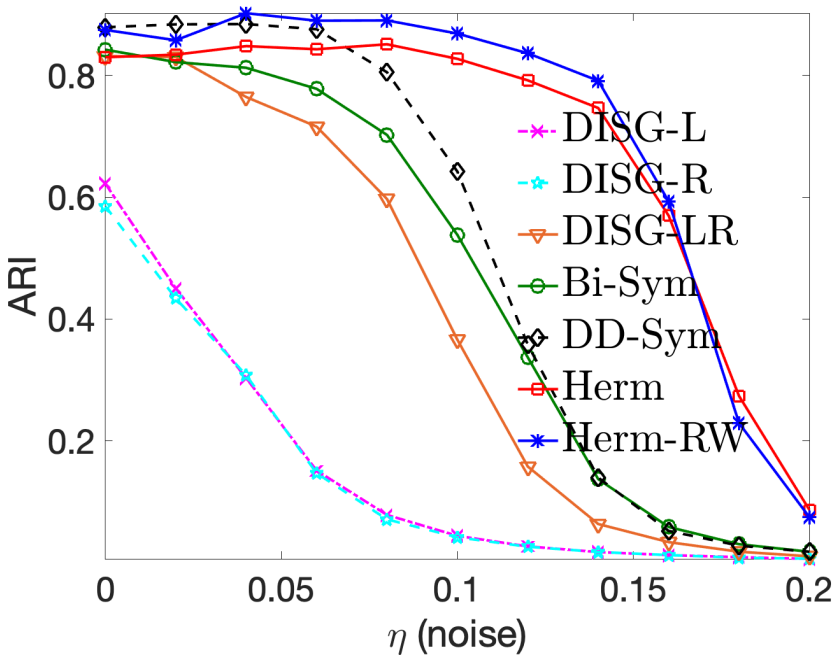
- **{Bi,DD}-Sym** V. Satuluri and S. Parthasarathy. *Symmetrizations for Clustering Directed Graphs*. In Proc. of the 14th ICEDT (2011)

Experiments - large k

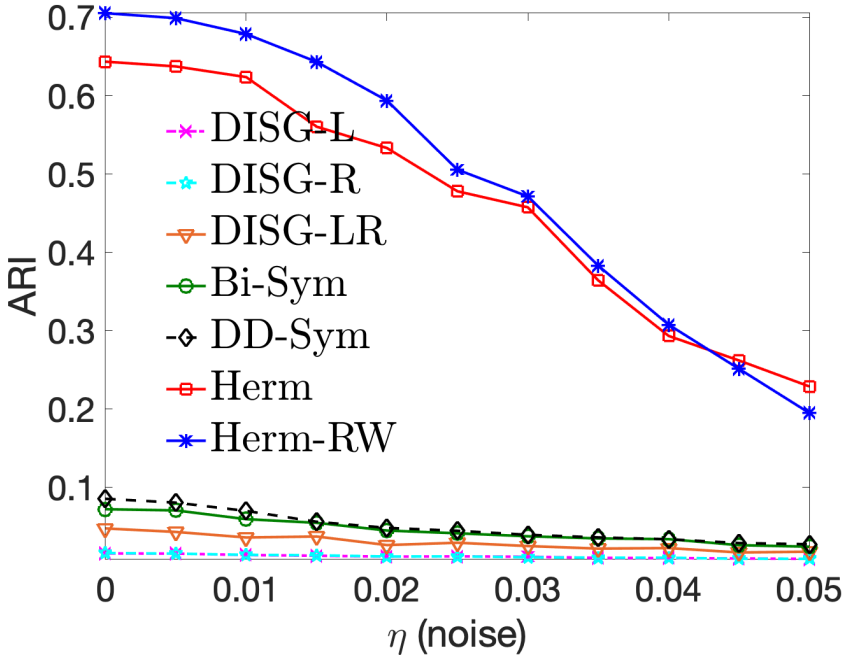


(a) $p = 0.02$

Experiments - large k



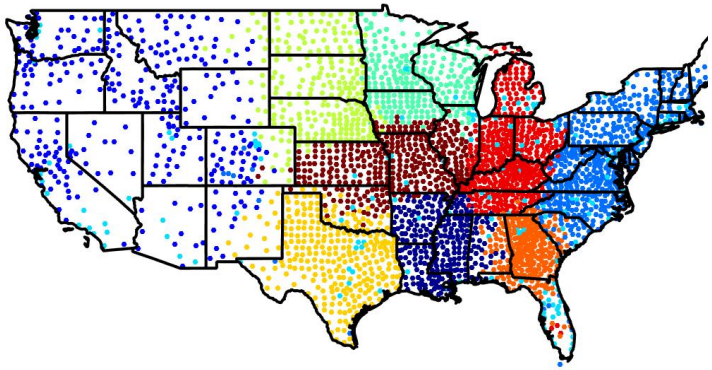
(a) $p = 0.02$



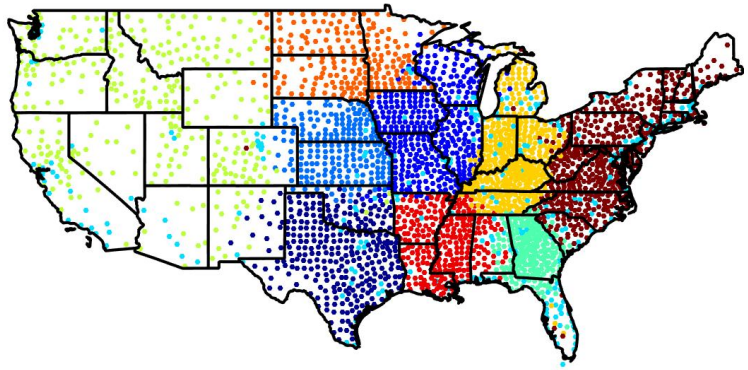
(b) $p = 0.01$

Figure: Recovery rates for the complete meta-graph in the DSBM with $k = 50$, $N = 5000$, two sparsity values p. Averaged over 10 runs.

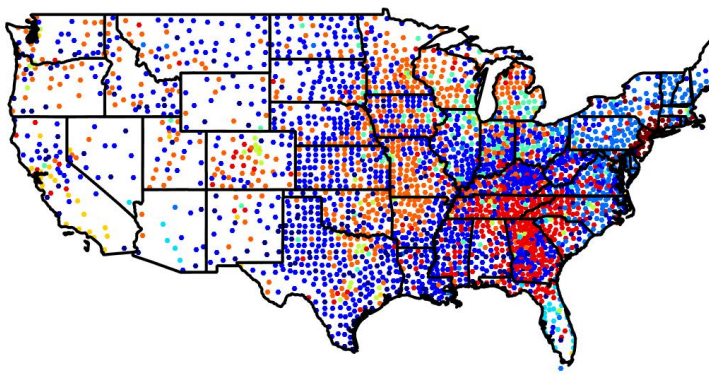
Clustering the US Migration Network



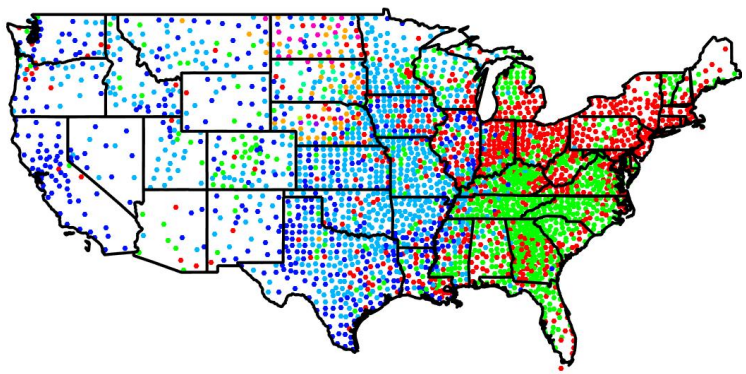
(a) DISGLR



(b) DD-SYM

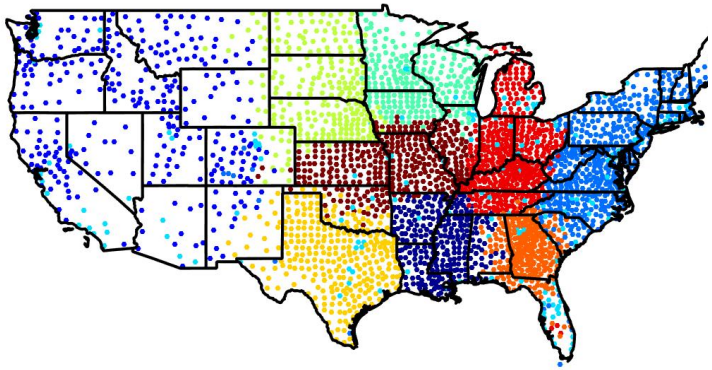


(c) HERM

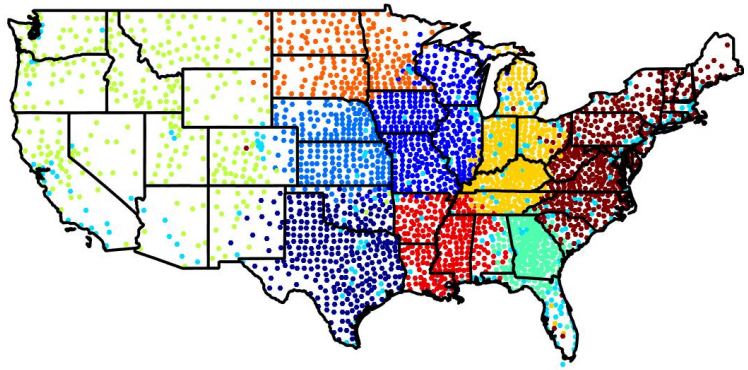


(d) HERM-RW

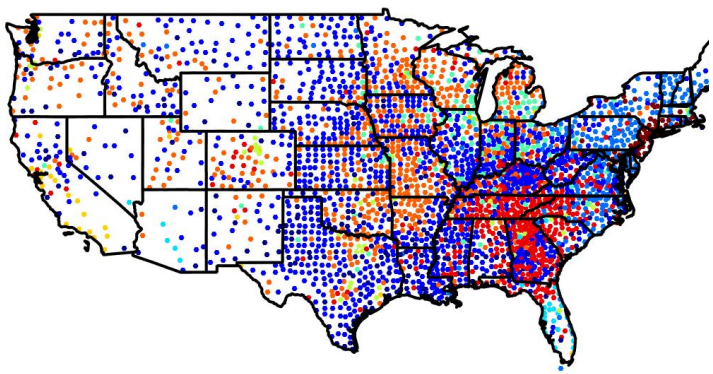
Clustering the US Migration Network



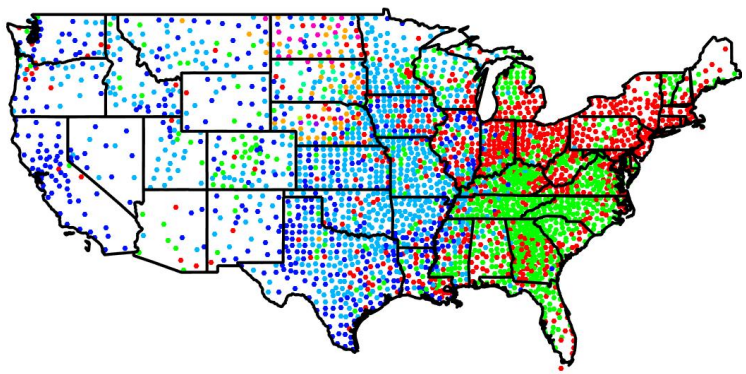
(e) DISGLR



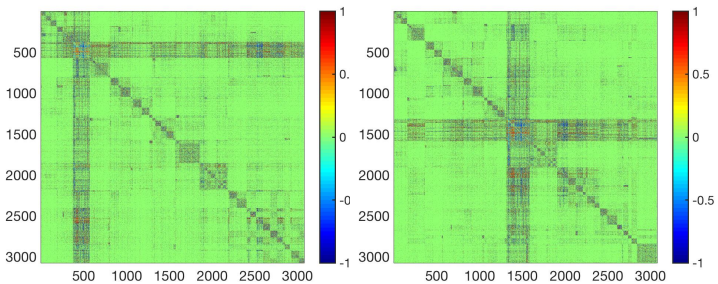
(f) DD-SYM



(g) HERM

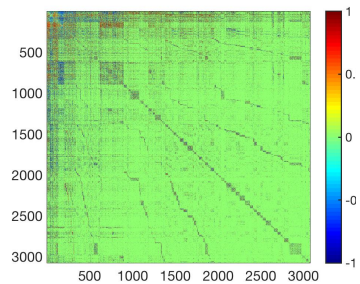


(h) HERM-RW

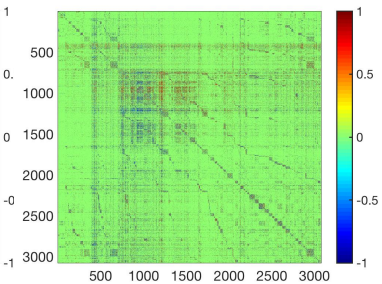


(a) DISGLR

(b) DD-SYM



(c) HERM



(d) HERM-RW

Cut Imbalance Ratio

Given a pair of clusters (X, Y), the Cut Imbalance ratio CI is defined by

$$\mathsf{CI}(X, Y) = \frac{w(X, Y)}{w(X, Y) + w(Y, X)}, \tag{25}$$

- $w(X, Y) = \sum_{j \in X, \ell \in Y} w(j, \ell)$ denotes the total weight of all edges flowing from X to Y
- indicates an imbalance for values closer to 0 or 1

Cut Imbalance Ratio

Given a pair of clusters (X, Y), the Cut Imbalance ratio CI is defined by

$$\mathsf{CI}(X, Y) = \frac{w(X, Y)}{w(X, Y) + w(Y, X)}, \tag{25}$$

- $w(X, Y) = \sum_{j \in X, \ell \in Y} w(j, \ell)$ denotes the total weight of all edges flowing from X to Y
- indicates an imbalance for values closer to 0 or 1
- normalization by size or volume, which we aim to maximize

$$\mathsf{CI}^{\text{size}}(X, Y) = |\mathsf{CI}(X, Y) - 0.5| \cdot \min\{|X|, |Y|\}, \tag{26}$$

Cut Imbalance Ratio

Given a pair of clusters (X, Y), the Cut Imbalance ratio CI is defined by

$$\mathrm{CI}(X, Y) = \frac{w(X, Y)}{w(X, Y) + w(Y, X)}, \tag{25}$$

- $w(X, Y) = \sum_{j \in X, \ell \in Y} w(j, \ell)$ denotes the total weight of all edges flowing from X to Y
- indicates an imbalance for values closer to 0 or 1
- normalization by size or volume, which we aim to maximize

$$\mathrm{CI}^{\mathrm{size}}(X, Y) = |\mathrm{CI}(X, Y) - 0.5| \cdot \min\{|X|, |Y|\}, \tag{26}$$

$$\mathrm{CI}^{\mathrm{vol}}(X, Y) = |\mathrm{CI}(X, Y) - 0.5| \cdot \min\{\mathrm{vol}(X), \mathrm{vol}(Y)\}, \tag{27}$$

- $\mathrm{vol}(X) = \sum_{j \in X} d_j^{\mathrm{in}} + d_j^{\mathrm{out}}$, denotes the sum of the total degrees of vertices in X. Ideally, we would like to maximize

Cut Imbalance Ratio

Given a pair of clusters (X, Y), the Cut Imbalance ratio CI is defined by

$$\mathrm{CI}(X, Y) = \frac{w(X, Y)}{w(X, Y) + w(Y, X)}, \tag{25}$$

- $w(X, Y) = \sum_{j \in X, \ell \in Y} w(j, \ell)$ denotes the total weight of all edges flowing from X to Y
- indicates an imbalance for values closer to 0 or 1
- normalization by size or volume, which we aim to maximize

$$\mathrm{CI}^{\mathrm{size}}(X, Y) = |\mathrm{CI}(X, Y) - 0.5| \cdot \min\{|X|, |Y|\}, \tag{26}$$

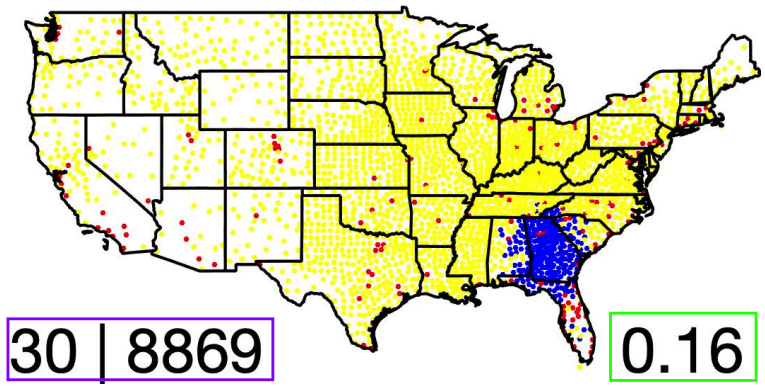
$$\mathrm{CI}^{\mathrm{vol}}(X, Y) = |\mathrm{CI}(X, Y) - 0.5| \cdot \min\{\mathrm{vol}(X), \mathrm{vol}(Y)\}, \tag{27}$$

- $\mathrm{vol}(X) = \sum_{j \in X} d_j^{\mathrm{in}} + d_j^{\mathrm{out}}$, denotes the sum of the total degrees of vertices in X. Ideally, we would like to maximize

$$\mathrm{TopCI}^{\mathrm{vol}} = \sum_{t=1}^{M} \mathrm{CI}^{\mathrm{vol}}(C_{j_t}, C_{\ell_t}) \tag{28}$$

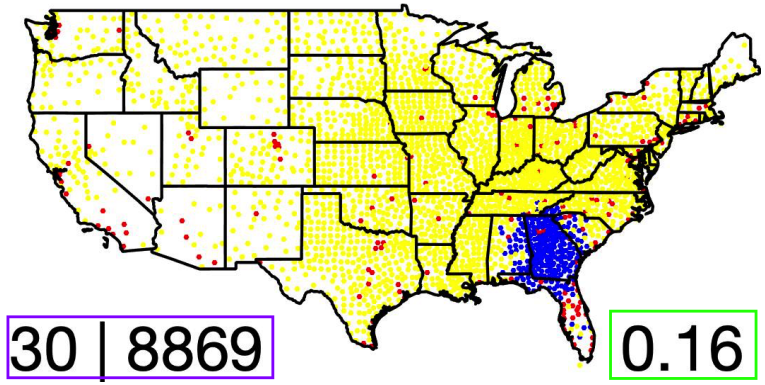
where (C_{j_t}, C_{ℓ_t}) denotes the t-th largest $\mathrm{CI}^{\mathrm{vol}}$ cut imbalance pair.

US Migration - top largest size-normalized cut imbalance pair

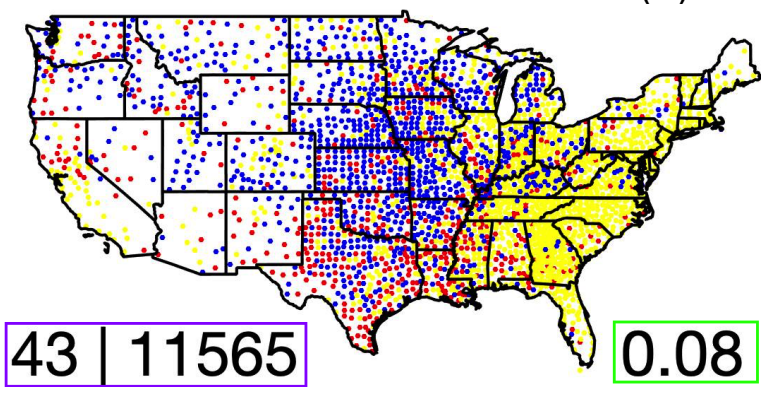


(a) DISGLR

US Migration - top largest size-normalized cut imbalance pair

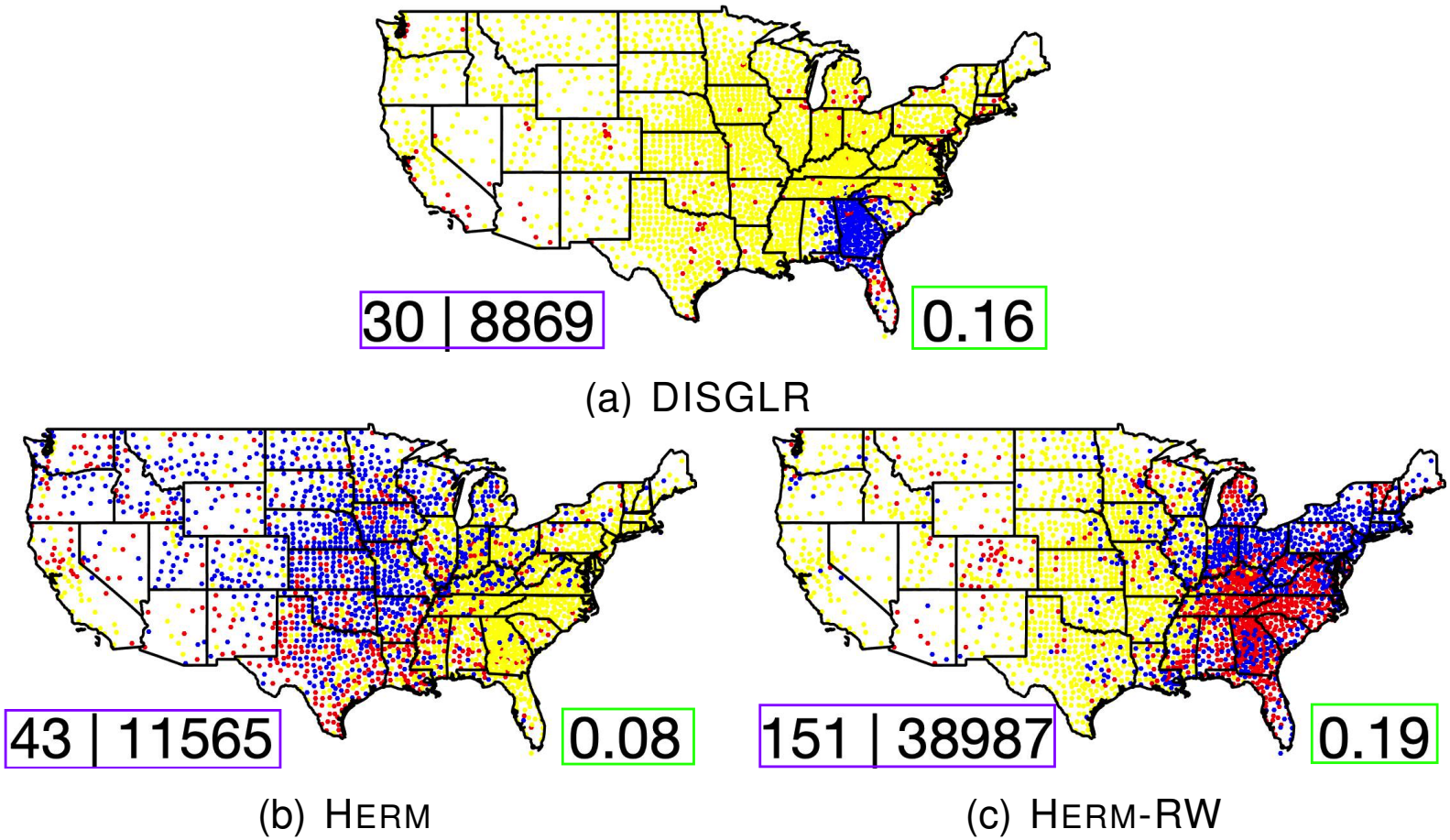


(a) DISGLR



(b) HERM

US Migration - top largest size-normalized cut imbalance pair



(a) DISGLR

(b) HERM

(c) HERM-RW

Figure: The top four largest size-normalized cut imbalance pairs for the US-MIGRATION-I data with $k = 10$ clusters. Red denotes the source cluster, and blue denotes the destination cluster. For each plot, the bottom left text contains the numerical values (rounded to nearest integer) of the normalized CI^{size} and CI^{vol} pairwise cut imbalance values (the higher the better), and the bottom right text contains the CI cut imbalance value in $[0, 1]$ (the farther from 0.5 the better).

Leaders and laggers in multivariate time series

- multivariate systems describing multiple quantities are thought to exhibit lead-lag relationships

Leaders and laggers in multivariate time series

- multivariate systems describing multiple quantities are thought to exhibit lead-lag relationships
- time series A leads time series B if A's past values are more strongly associated with B's future values than A's future values are with B's past values

Leaders and laggers in multivariate time series

- multivariate systems describing multiple quantities are thought to exhibit lead-lag relationships
- time series A leads time series B if A's past values are more strongly associated with B's future values than A's future values are with B's past values
- lagged relationships often encountered in natural physical systems
- of interest in fields such as finance, economics, earth science, biology

Leaders and laggers in multivariate time series

- multivariate systems describing multiple quantities are thought to exhibit lead-lag relationships
- time series A leads time series B if A's past values are more strongly associated with B's future values than A's future values are with B's past values
- lagged relationships often encountered in natural physical systems
- of interest in fields such as finance, economics, earth science, biology
- one time series has a delayed response
  - to the other series,
  - to a common factor/stimulus that affects both series

Leaders and laggers in multivariate time series

- multivariate systems describing multiple quantities are thought to exhibit lead-lag relationships
- time series A leads time series B if A's past values are more strongly associated with B's future values than A's future values are with B's past values
- lagged relationships often encountered in natural physical systems
- of interest in fields such as finance, economics, earth science, biology
- one time series has a delayed response
  - to the other series,
  - to a common factor/stimulus that affects both series
- clustering/ranking/denoising arises from lagged relationships

Leaders and laggers in multivariate time series

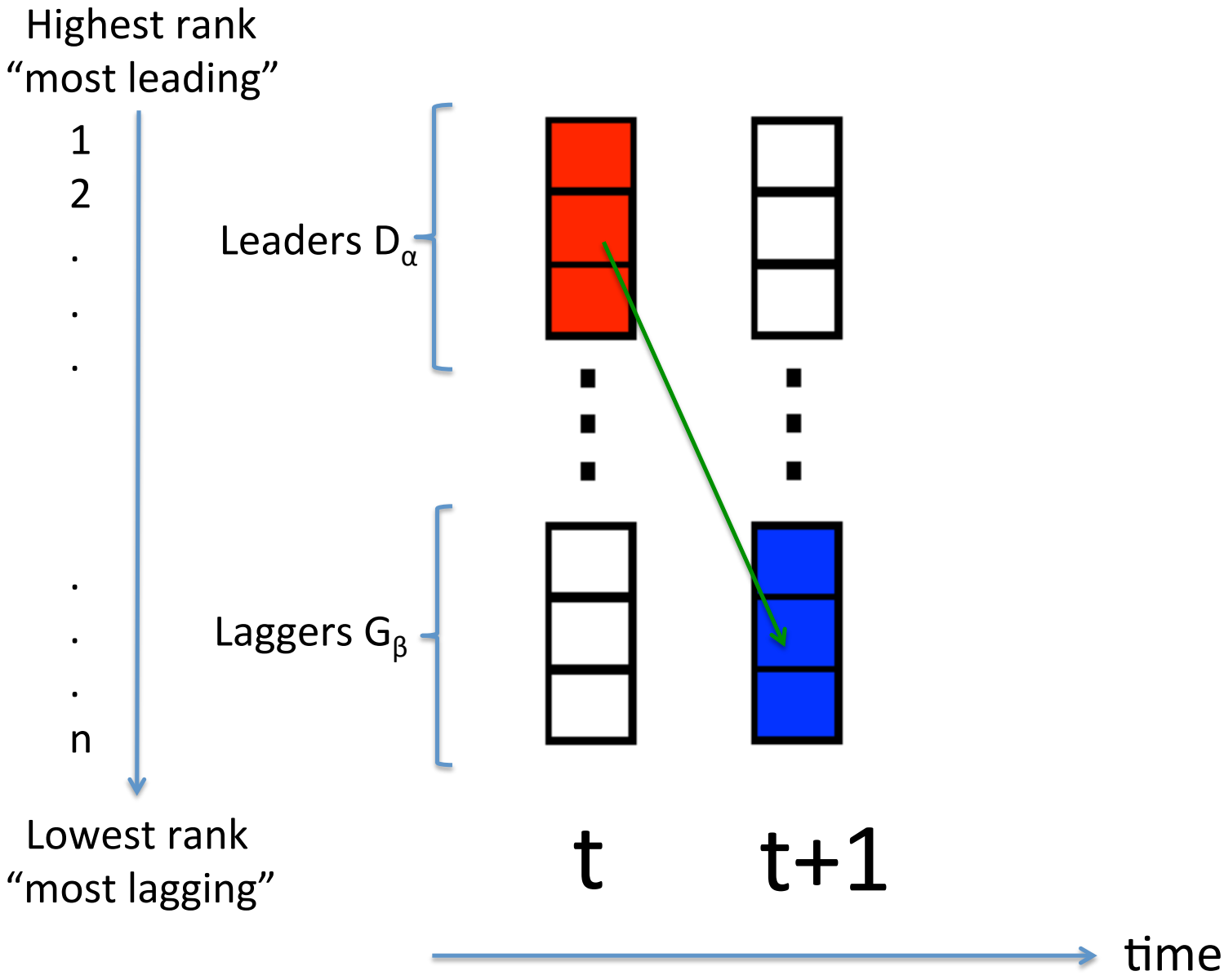
- multivariate systems describing multiple quantities are thought to exhibit lead-lag relationships
- time series A leads time series B if A's past values are more strongly associated with B's future values than A's future values are with B's past values
- lagged relationships often encountered in natural physical systems
- of interest in fields such as finance, economics, earth science, biology
- one time series has a delayed response
  - to the other series,
  - to a common factor/stimulus that affects both series
- clustering/ranking/denoising arises from lagged relationships
- the return of instrument i on day t may influence the behavior of instrument j on day $t + 3$ (i *leads* j by 3 units of time)

Leaders and laggers in multivariate time series

- multivariate systems describing multiple quantities are thought to exhibit lead-lag relationships
- time series A leads time series B if A's past values are more strongly associated with B's future values than A's future values are with B's past values
- lagged relationships often encountered in natural physical systems
- of interest in fields such as finance, economics, earth science, biology
- one time series has a delayed response
  - to the other series,
  - to a common factor/stimulus that affects both series
- clustering/ranking/denoising arises from lagged relationships
- the return of instrument i on day t may influence the behavior of instrument j on day $t + 3$ (i *leads* j by 3 units of time)
- such pairwise comparisons are very noisy and inconsistent
  - i leads j by 3 time units, j leads k by 2 units, but i does not lead k by 5 units
- give rise to lead-lag networks from multivariate time series

Leaders and laggers in multivariate time series

- multivariate systems describing multiple quantities are thought to exhibit lead-lag relationships
- time series A leads time series B if A's past values are more strongly associated with B's future values than A's future values are with B's past values
- lagged relationships often encountered in natural physical systems
- of interest in fields such as finance, economics, earth science, biology
- one time series has a delayed response
  - to the other series,
  - to a common factor/stimulus that affects both series
- clustering/ranking/denoising arises from lagged relationships
- the return of instrument i on day t may influence the behavior of instrument j on day $t + 3$ (i *leads* j by 3 units of time)
- such pairwise comparisons are very noisy and inconsistent
  - i leads j by 3 time units, j leads k by 2 units, but i does not lead k by 5 units
- give rise to lead-lag networks from multivariate time series
  - compute rankings, predict the laggers catching up
  - cluster the network, and infer leading and lagging clusters, and leverage for prediction task


Highest rank
“most leading”
1
2
.
.
.
Leaders D_α
.
.
.
n
Laggers G_β
Lowest rank
“most lagging”
t
t+1
time

Why lead-lag relationships?

- Stock returns are difficult to predict due to low signal to noise ratio
- Changes in stock prices of some firms tend to follow that of other firms, and this relationship between stock prices is often referred to as *lead-lag* relationships
- Detecting lead-lag relationships is not straightforward

It would be useful if one could:

1. Detect lead-lag relationships
2. Use lead-lag relationships to predict stock returns
3. Employ these relationships to build portfolios

Outline of our lead-lag project

1. We propose a data-driven method to detect lead-lag relationships in stock returns
   - We do not assume a linear relationship between leaders and followers

Outline of our lead-lag project

1. We propose a data-driven method to detect lead-lag relationships in stock returns
   - We do not assume a linear relationship between leaders and followers

2. We construct a portfolio of assets with state-of-the-art ranking and clustering methods that use the Lévy–area and cross-correlation between asset returns

Outline of our lead-lag project

1. We propose a data-driven method to detect lead-lag relationships in stock returns
   - We do not assume a linear relationship between leaders and followers

2. We construct a portfolio of assets with state-of-the-art ranking and clustering methods that use the Lévy–area and cross-correlation between asset returns

3. We find economically significant lead-lag relationships that outperform all previous benchmarks in the literature,

Outline of our lead-lag project

1. We propose a data-driven method to detect lead-lag relationships in stock returns
   - We do not assume a linear relationship between leaders and followers

2. We construct a portfolio of assets with state-of-the-art ranking and clustering methods that use the Lévy–area and cross-correlation between asset returns

3. We find economically significant lead-lag relationships that outperform all previous benchmarks in the literature,

4. There is little overlap in portfolio composition between the lead-lag portfolio we construct and those constructed in previous studies

Outline of our lead-lag project

1. We propose a data-driven method to detect lead-lag relationships in stock returns
   - We do not assume a linear relationship between leaders and followers

2. We construct a portfolio of assets with state-of-the-art ranking and clustering methods that use the Lévy–area and cross-correlation between asset returns

3. We find economically significant lead-lag relationships that outperform all previous benchmarks in the literature,

4. There is little overlap in portfolio composition between the lead-lag portfolio we construct and those constructed in previous studies

5. We find that the portfolios based on lead-lag relationships are sensitive to rebalancing frequencies. Portfolios rebalanced once a day consistently outperform the bidaily, weekly, bi-weekly, tri-weekly, and monthly rebalanced portfolios

Lead-lag literature in finance

Lead-lag relationships are intensively studied in the literature of return predictability

- Lo and MacKinlay (1990): Firms with large market cap lead firms with small market cap
- Badrinath et al. (1995): Firms with high institutional ownership lead firms with low institutional ownership
- Chordia and Swaminathan (2000): High volume/turnover stocks lead low volume/turnover stocks
- Hou (2007): large cap stocks lead small cap stocks in each industry
- Brennan et al. (2015): High investment coverage stocks lead low coverage stocks
- Parsons et al. (2020): stocks who are not in the same sector but have the same headquarter show lead-lag relationships, with large cap ones leading small cap ones
- Huang et al. (2022): stocks with frequent, gradual price updates lead those with infrequent, dramatic price updates.

Related literature – source of lead-lag

- previous literature: leaders & followers are often pre-determined by economic heuristics (e.g. market cap, volume etc.)

Related literature – source of lead-lag

- previous literature: leaders & followers are often pre-determined by economic heuristics (e.g. market cap, volume etc.)
- literature mostly attributes the source of lead-lag relationship to the *slow information diffusion hypothesis*
  - information spreads gradually throughout financial markets, resulting in delayed price adjustments

Related literature – source of lead-lag

- previous literature: leaders & followers are often pre-determined by economic heuristics (e.g. market cap, volume etc.)
- literature mostly attributes the source of lead-lag relationship to the *slow information diffusion hypothesis*
  - information spreads gradually throughout financial markets, resulting in delayed price adjustments
- some assets receive information faster, and others take longer to receive information and to incorporate the price change

Related literature – source of lead-lag

- previous literature: leaders & followers are often pre-determined by economic heuristics (e.g. market cap, volume etc.)
- literature mostly attributes the source of lead-lag relationship to the *slow information diffusion hypothesis*
  - information spreads gradually throughout financial markets, resulting in delayed price adjustments
- some assets receive information faster, and others take longer to receive information and to incorporate the price change
- if the slow information diffusion hypothesis holds, it is natural to
  - extend beyond pre-determined leader-follower identities
  - use data-driven methods to better understand dynamics of information flow in financial markets.

Pairwise lead-lag detection methodology

- finance & econometrics literature: typical methods in causality relationship detection rely on the Granger causality test (Granger 1969)
  - linear regression test that assesses if the inclusion of the lagged values of one series in a linear regression model significantly improves the fit and predictive power of the model compared to the original model.
  - shortfalls: assumes a linear relationship between the leader and the follower + stationarity

Pairwise lead-lag detection methodology

- finance & econometrics literature: typical methods in causality relationship detection rely on the Granger causality test (Granger 1969)
  - linear regression test that assesses if the inclusion of the lagged values of one series in a linear regression model significantly improves the fit and predictive power of the model compared to the original model.
  - shortfalls: assumes a linear relationship between the leader and the follower + stationarity
- in machine learning and statistics, previous works have used cross-correlation based methods to detect lead-lag.
  - e.g. Chen (1992) studies a lead-lag between the spot and future market.
  - we use cross-correlation as one of the methods to detect lead-lag as well.

Pairwise lead-lag detection methodology

- finance & econometrics literature: typical methods in causality relationship detection rely on the Granger causality test (Granger 1969)
  - linear regression test that assesses if the inclusion of the lagged values of one series in a linear regression model significantly improves the fit and predictive power of the model compared to the original model.
  - shortfalls: assumes a linear relationship between the leader and the follower + stationarity
- in machine learning and statistics, previous works have used cross-correlation based methods to detect lead-lag.
  - e.g. Chen (1992) studies a lead-lag between the spot and future market.
  - we use cross-correlation as one of the methods to detect lead-lag as well.
- we also introduce a method based on signatures to handle nonlinearity in the data.
- just a means to an end in terms of detecting higher-order structure

Data-driven lead-lag detection pipeline

- we propose a data-driven pipeline aiming to capture the fast-changing dynamic inherent in lead-lag relationships

Data-driven lead-lag detection pipeline

- we propose a data-driven pipeline aiming to capture the fast-changing dynamic inherent in lead-lag relationships
- our methods are capable of detecting and verifying lead-lag relationships effectively and deliver superior portfolio performances

Data-driven lead-lag detection pipeline

- we propose a data-driven pipeline aiming to capture the fast-changing dynamic inherent in lead-lag relationships
- our methods are capable of detecting and verifying lead-lag relationships effectively and deliver superior portfolio performances

Outline:

1. measure pairwise lead-lag relationships

Data-driven lead-lag detection pipeline

- we propose a data-driven pipeline aiming to capture the fast-changing dynamic inherent in lead-lag relationships
- our methods are capable of detecting and verifying lead-lag relationships effectively and deliver superior portfolio performances

Outline:

1. measure pairwise lead-lag relationships
2. extrapolate pairwise relationships to higher-order relationships

Data-driven lead-lag detection pipeline

- we propose a data-driven pipeline aiming to capture the fast-changing dynamic inherent in lead-lag relationships
- our methods are capable of detecting and verifying lead-lag relationships effectively and deliver superior portfolio performances

Outline:

1. measure pairwise lead-lag relationships
2. extrapolate pairwise relationships to higher-order relationships
   - (A) MetaCluster: build cluster-driven portfolios that lead-lag each other, or

Data-driven lead-lag detection pipeline

- we propose a data-driven pipeline aiming to capture the fast-changing dynamic inherent in lead-lag relationships
- our methods are capable of detecting and verifying lead-lag relationships effectively and deliver superior portfolio performances

Outline:

1. measure pairwise lead-lag relationships
2. extrapolate pairwise relationships to higher-order relationships
   - (A) MetaCluster: build cluster-driven portfolios that lead-lag each other, or
   - (B) GlobalRank: extract ranking-based ”global" leaders and laggers/followers, or

Data-driven lead-lag detection pipeline

- we propose a data-driven pipeline aiming to capture the fast-changing dynamic inherent in lead-lag relationships
- our methods are capable of detecting and verifying lead-lag relationships effectively and deliver superior portfolio performances

Outline:

1. measure pairwise lead-lag relationships
2. extrapolate pairwise relationships to higher-order relationships
   - (A) MetaCluster: build cluster-driven portfolios that lead-lag each other, or
   - (B) GlobalRank: extract ranking-based ”global" leaders and laggers/followers, or
   - (C) ClusterRank: combine clustering & ranking for lead-lag extraction

Data-driven lead-lag detection pipeline

- we propose a data-driven pipeline aiming to capture the fast-changing dynamic inherent in lead-lag relationships
- our methods are capable of detecting and verifying lead-lag relationships effectively and deliver superior portfolio performances

Outline:

1. measure pairwise lead-lag relationships
2. extrapolate pairwise relationships to higher-order relationships
   - (A) MetaCluster: build cluster-driven portfolios that lead-lag each other, or
   - (B) GlobalRank: extract ranking-based ”global" leaders and laggers/followers, or
   - (C) ClusterRank: combine clustering & ranking for lead-lag extraction
3. construct a portfolio to test/evaluate/exploit the uncovered lead-lag relationship.

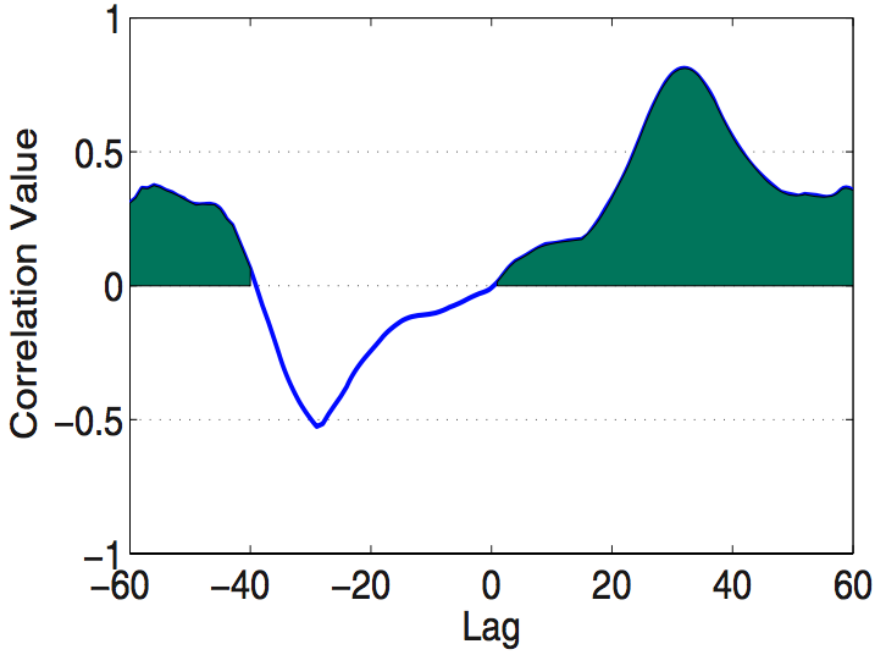
Data-driven lead-lag detection pipeline

- we propose a data-driven pipeline aiming to capture the fast-changing dynamic inherent in lead-lag relationships
- our methods are capable of detecting and verifying lead-lag relationships effectively and deliver superior portfolio performances

Outline:

1. measure pairwise lead-lag relationships
2. extrapolate pairwise relationships to higher-order relationships
   - (A) MetaCluster: build cluster-driven portfolios that lead-lag each other, or
   - (B) GlobalRank: extract ranking-based ”global" leaders and laggers/followers, or
   - (C) ClusterRank: combine clustering & ranking for lead-lag extraction
3. construct a portfolio to test/evaluate/exploit the uncovered lead-lag relationship.

Cross-correlations and the lead-lag matrix

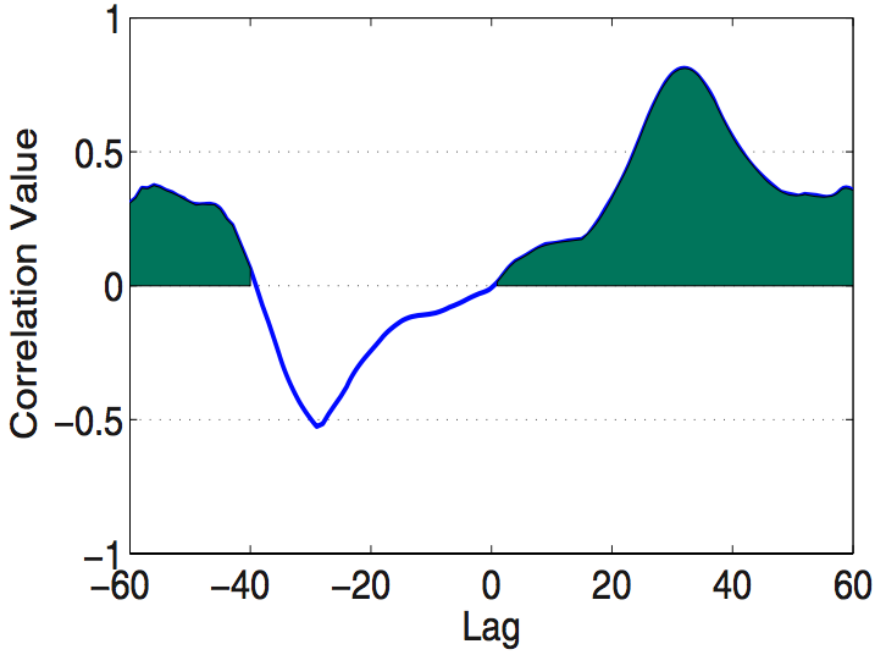


Wu et al, 2010

Options for building the $n \times n$ pairwise comparison matrix:

1. C_{ij}: lag that maximizes the cross-correlation

Cross-correlations and the lead-lag matrix

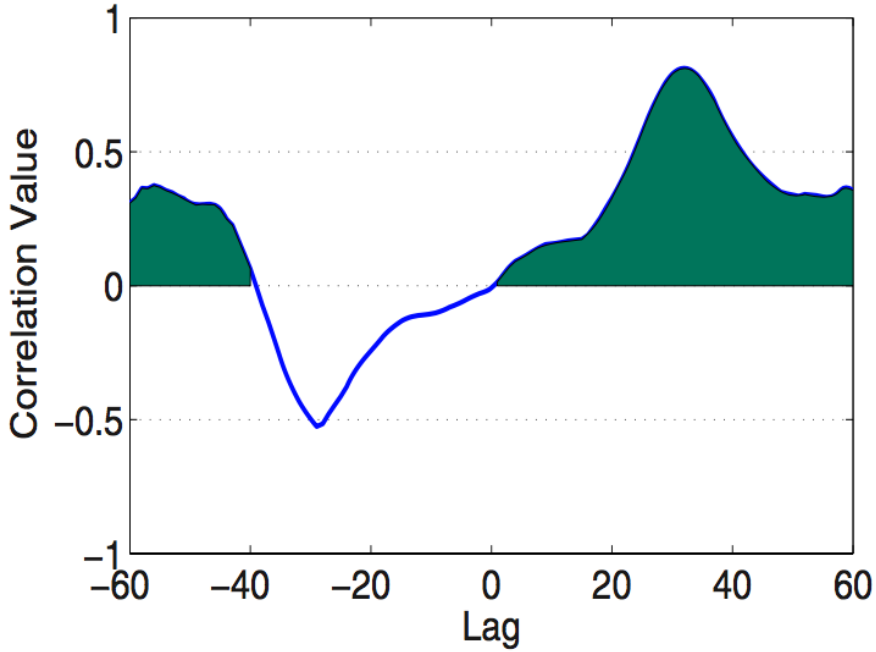


Wu et al, 2010

Options for building the $n \times n$ pairwise comparison matrix:

1. C_{ij}: lag that maximizes the cross-correlation
2. C_{ij}: $\pm \max\{\text{avg. corr. of +ve lags}, \text{avg. corr. of -ve lags}\}$

Cross-correlations and the lead-lag matrix



Wu et al, 2010

Options for building the $n \times n$ pairwise comparison matrix:

1. C_{ij}: lag that maximizes the cross-correlation
2. C_{ij}: $\pm \max\{\text{avg. corr. of +ve lags}, \text{avg. corr. of -ve lags}\}$
3. C_{ij}: second-order signatures of the two time series

$$C_{ij}(t-m,t) = \iint_{t-m<u<v<t} dX_i(u)\, dX_j(v) - dX_j(u)\, dX_i(v)$$

Lead-lag detection methodology/framework

In a simple linear model, consider the regression

$$r_{i,t} = \beta_\ell r_{j,t-\ell} + \epsilon_t \tag{29}$$

i, j represent different assets from a universe of n assets, and t indexes time.

- for each pair of assets (i, j), identify one of the two as the leader and the other as the follower
- ideally, we want to design an algorithm such that by assigning a score C_{ij} to each asset pair (i, j), the score satisfies two conditions:
  1. the sign of the score determines the direction of the lead-lag relationship (i.e. the sign of ℓ)
  2. the magnitude of the score determines the strength of the relationship. (i.e. proportional to β_ℓ)
- yields an $n \times n$ matrix denoted as the lead-lag matrix C (comparison matrix), on which we can perform ranking/clustering algorithms.

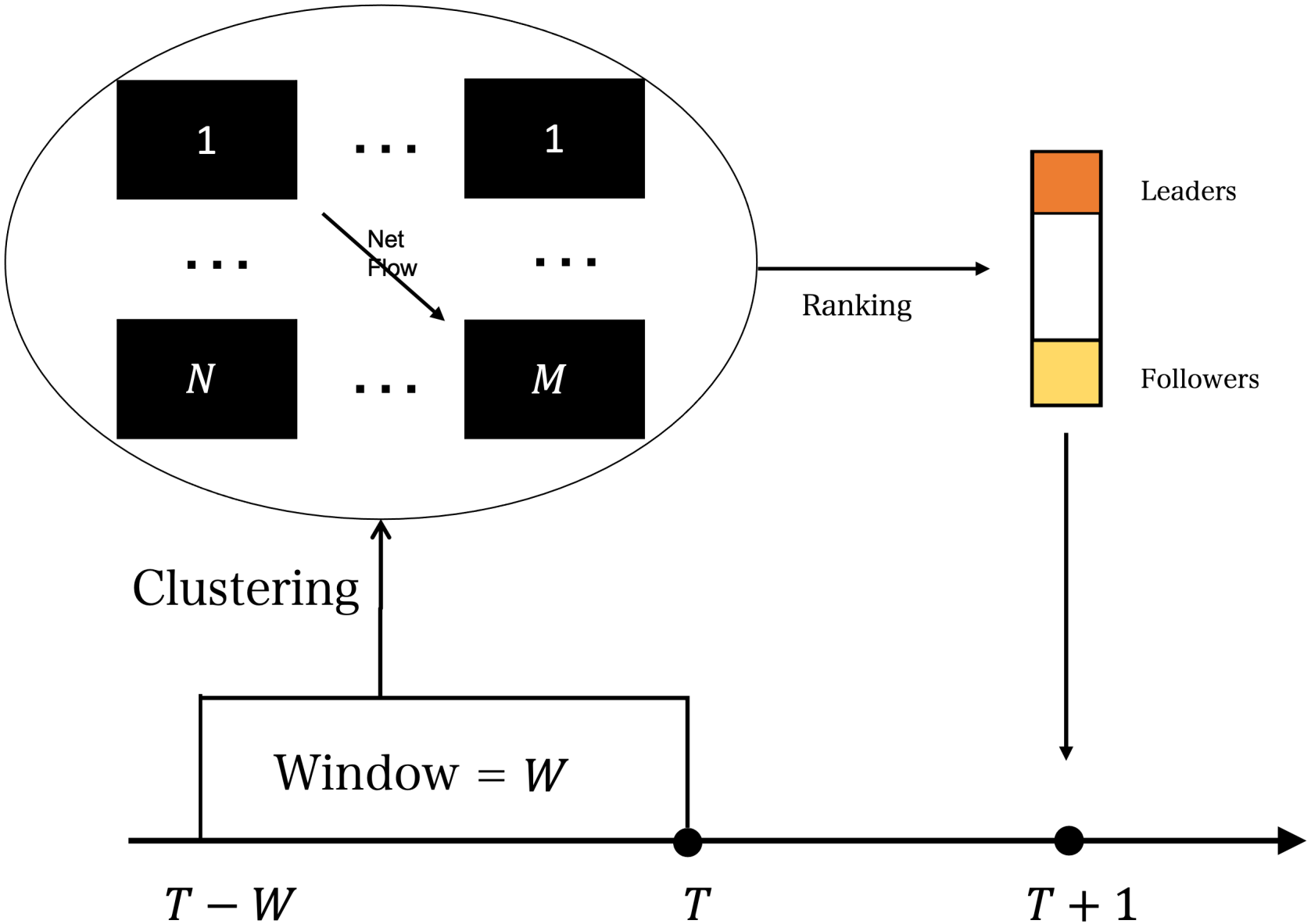
Data-driven lead-lag detection pipeline

- we propose a data-driven pipeline aiming to capture the fast-changing dynamic inherent in lead-lag relationships
- our methods are capable of detecting and verifying lead-lag relationships effectively and deliver superior portfolio performances

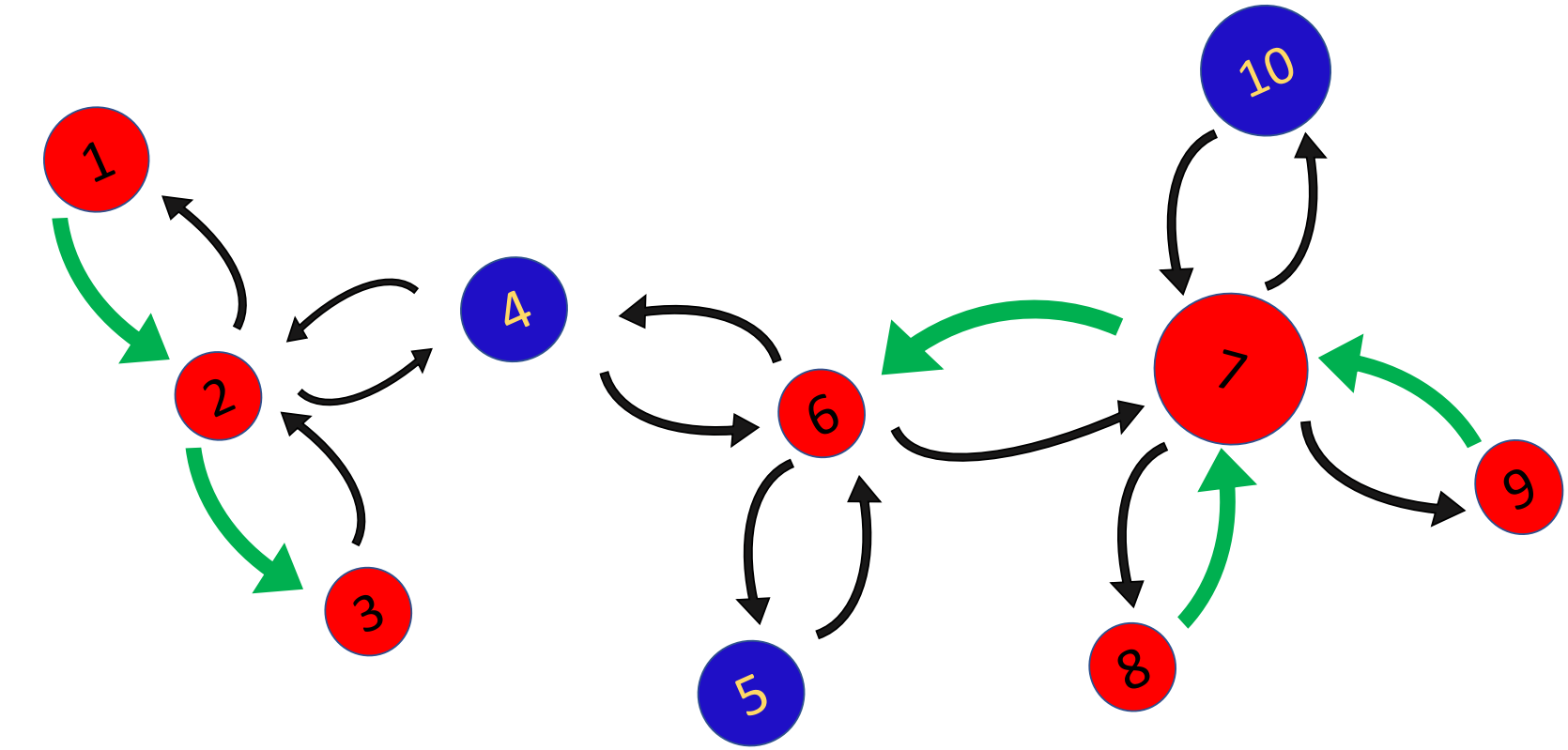
Outline:

1. measure pairwise lead-lag relationships
2. extrapolate pairwise relationships to higher-order relationships
   - (A) MetaCluster: build cluster-driven portfolios that lead-lag each other
   - (B) GlobalRank: extract ranking-based ”global" leaders and laggers/followers, or
   - (C) ClusterRank: combine clustering & ranking for lead-lag extraction
3. construct a portfolio to test/evaluate/exploit the uncovered lead-lag relationship.

MetaClusters diagram



Uncover underlying directed meta-graph structure



- Bennett, C. and Reinert, *"Detection and clustering of lead-lag networks for multivariate time series with an application to financial markets"*, Machine Learning (2022)

Data description

- universe of 5325 NYSE equities spanning from 04-01-2000 to 31-12-2019
- Wharton's CRSP database – restricting our attention to equities trading on the same exchange to avoid spurious lead-lag effects due to non-synchronous trading
- daily closing prices from which we compute daily log-returns
- subset to symbols with at least 2.5 years' worth of non-missing data
- subset the largest 500 equities in average daily traded dollar volume
- result in a data set of 434 equities
- reduces the risk of spurious lead-lag effects due to non-synchronous trading

Cluster meta-flow matrix & cluster leadingness metric

- total *net flow* between any two clusters is given by the net of the normalized weights between all edges directed from one cluster to another

Cluster meta-flow matrix & cluster leadingness metric

- total *net flow* between any two clusters is given by the net of the normalized weights between all edges directed from one cluster to another
- encode this in the skew-symmetric *meta-flow* matrix F

$$F_{ij} = \frac{1}{|C_i|\,|C_j|} \sum_{u \in C_i, v \in C_j} [A_{uv} - A_{vu}],$$

- C_i denotes all nodes in cluster i, and $i, j \in \{1, \ldots, k\}$, $i \neq j$

Cluster meta-flow matrix & cluster leadingness metric

- total *net flow* between any two clusters is given by the net of the normalized weights between all edges directed from one cluster to another
- encode this in the skew-symmetric *meta-flow* matrix F

$$F_{ij} = \frac{1}{|C_i|\,|C_j|} \sum_{u \in C_i, v \in C_j} [A_{uv} - A_{vu}],$$

- C_i denotes all nodes in cluster i, and $i, j \in \{1, \ldots, k\}$, $i \neq j$
- the diagonal of F consists of zeros: $F_{ii} = 0, \ \forall i \in 1, \ldots, k$

Cluster meta-flow matrix & cluster leadingness metric

- total *net flow* between any two clusters is given by the net of the normalized weights between all edges directed from one cluster to another
- encode this in the skew-symmetric *meta-flow* matrix F

$$F_{ij} = \frac{1}{|C_i|\,|C_j|} \sum_{u \in C_i, v \in C_j} [A_{uv} - A_{vu}],$$

- C_i denotes all nodes in cluster i, and $i, j \in \{1, \ldots, k\}$, $i \neq j$
- the diagonal of F consists of zeros: $F_{ii} = 0$, $\forall\, i \in 1, \ldots, k$
- define a metric for the *leadingness* of each cluster $i \in \{1, \ldots, k\}$

$$L(i) := \frac{1}{|C_i|} \sum_{u \in C_i, v \in \{1,\ldots,n\}} [A_{uv} - A_{vu}]. \tag{30}$$

Thus, $L(i)$ averages the row-sums of the skew-symmetric matrix $A - A^T$ for nodes within the cluster C_i;

Cluster meta-flow matrix & cluster leadingness metric

- total *net flow* between any two clusters is given by the net of the normalized weights between all edges directed from one cluster to another
- encode this in the skew-symmetric *meta-flow* matrix F

$$F_{ij} = \frac{1}{|C_i|\,|C_j|} \sum_{u \in C_i, v \in C_j} [A_{uv} - A_{vu}],$$

- C_i denotes all nodes in cluster i, and $i, j \in \{1, \ldots, k\}$, $i \neq j$
- the diagonal of F consists of zeros: $F_{ii} = 0$, $\forall\, i \in 1, \ldots, k$
- define a metric for the *leadingness* of each cluster $i \in \{1, \ldots, k\}$

$$L(i) := \frac{1}{|C_i|} \sum_{u \in C_i, v \in \{1,\ldots,n\}} [A_{uv} - A_{vu}]. \tag{30}$$

Thus, $L(i)$ averages the row-sums of the skew-symmetric matrix $A - A^T$ for nodes within the cluster C_i;

- the row-sums of the lead-lag matrix are measure of the total tendency of the equity corresponding to the row to be a leader

Cluster meta-flow matrix & cluster leadingness metric

- total *net flow* between any two clusters is given by the net of the normalized weights between all edges directed from one cluster to another
- encode this in the skew-symmetric *meta-flow* matrix F

$$F_{ij} = \frac{1}{|C_i|\,|C_j|} \sum_{u \in C_i, v \in C_j} [A_{uv} - A_{vu}],$$

- C_i denotes all nodes in cluster i, and $i, j \in \{1, \ldots, k\}$, $i \neq j$
- the diagonal of F consists of zeros: $F_{ii} = 0,\ \forall\, i \in 1, \ldots, k$
- define a metric for the *leadingness* of each cluster $i \in \{1, \ldots, k\}$

$$L(i) := \frac{1}{|C_i|} \sum_{u \in C_i, v \in \{1,\ldots,n\}} [A_{uv} - A_{vu}]. \tag{30}$$

Thus, $L(i)$ averages the row-sums of the skew-symmetric matrix $A - A^T$ for nodes within the cluster C_i;

- the row-sums of the lead-lag matrix are measure of the total tendency of the equity corresponding to the row to be a leader
- obtain a ranking of the clusters from
  - the most leading cluster (largest row-sum value) (label 0)
  - to the most lagging cluster (smallest row-sum value) (label $k - 1$)

From the stock lead-lag matrix to the cluster meta-flow matrix

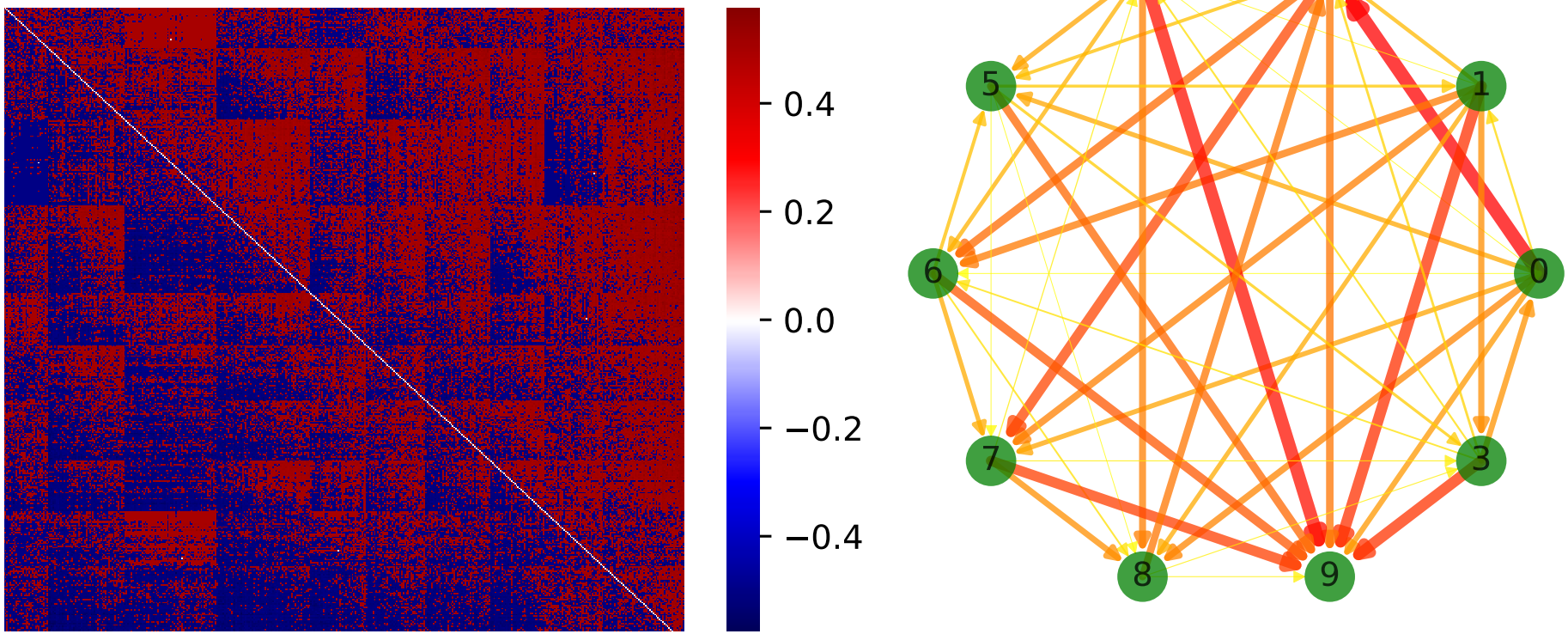


Figure: Left: Heatmap of the double-sorted lead-lag $n \times n$ matrix $A - A^T$. The rows and columns of the matrix index the $n = 434$ equities, and are categorised by cluster membership (labelled by the leadingness metric. Within each cluster, we sort the equities by their respective row-sum in $A - A^T$, a proxy for their individual leadingness. Right: Meta-flow network for Hermitian RW clusters; clusters are represented by nodes and larger edge weights are depicted by bolder colours and thicker lines. Cluster 0: most leading; Cluster 9 most lagging.

Clusters vs GICS

Retail	90
Manufacturing	67
Construction	66
Mining	58
Trans., Util. & other	54
Fin., Ins. & RE	46
Wholesale	43
Services	9
Agri., Forest. & Fish.	1

Table: Number of equities in each SIC industry sector.

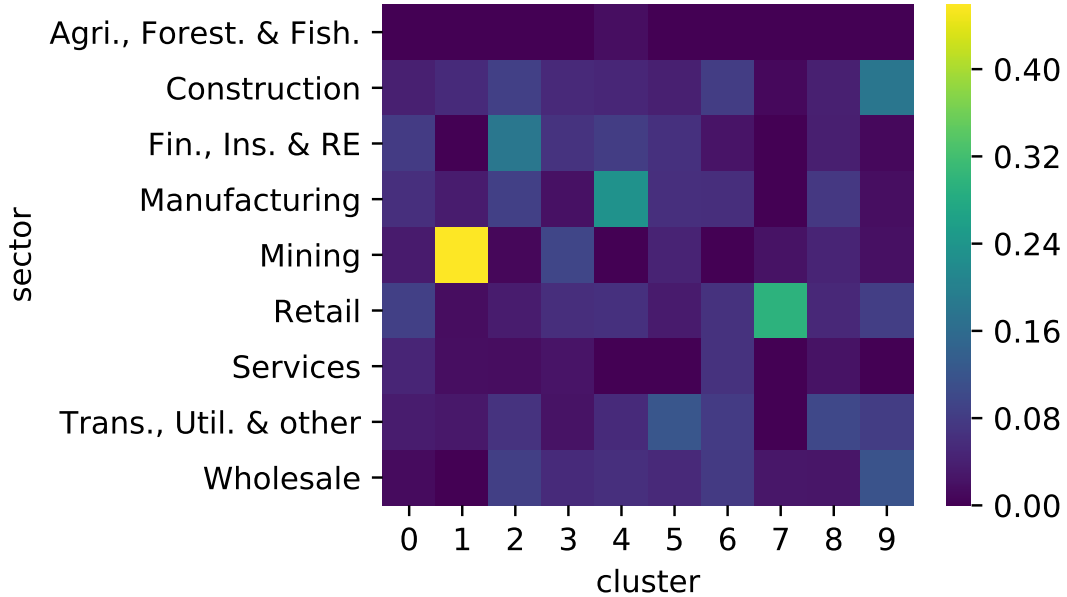


Figure: The Jaccard similarity coefficient between the Hermitian RW clusters and industry clusters. Cluster 0 is most leading; cluster 9 is most lagging.

Clusters vs GICS

Retail	90
Manufacturing	67
Construction	66
Mining	58
Trans., Util. & other	54
Fin., Ins. & RE	46
Wholesale	43
Services	9
Agri., Forest. & Fish.	1

Table: Number of equities in each SIC industry sector.

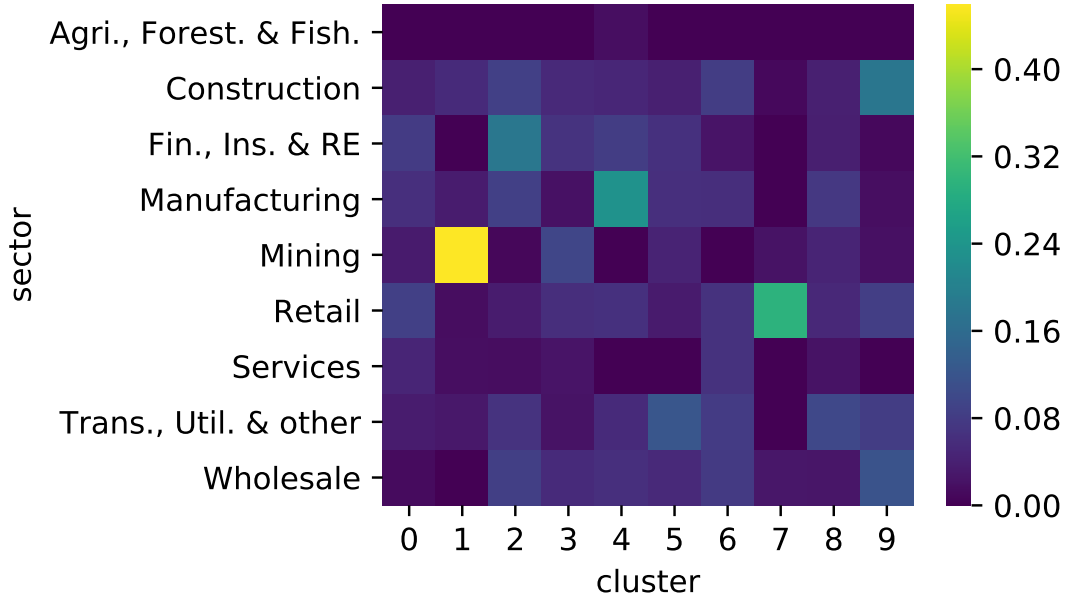


Figure: The Jaccard similarity coefficient between the Hermitian RW clusters and industry clusters. Cluster 0 is most leading; cluster 9 is most lagging.

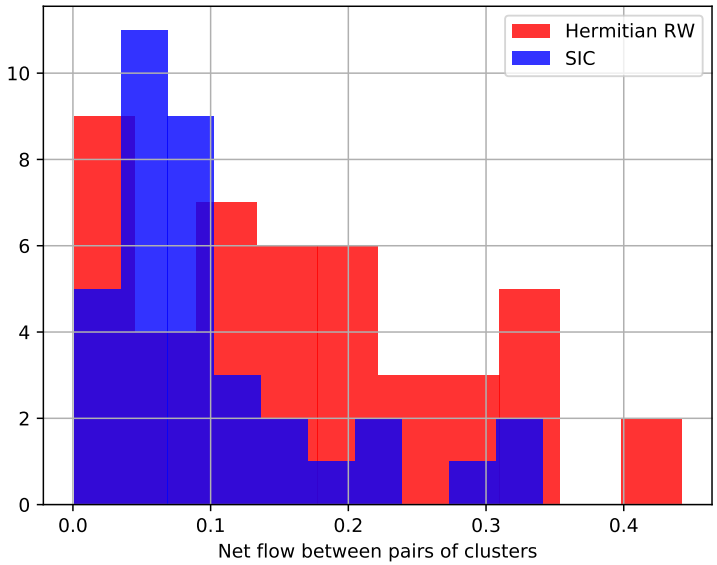
- test for a statistically significant time dependence in daily returns using a permutation test on the spectrum of the Hermitian adj. mtx. $\tilde{A} = i(A - A^T)$
- since lead-lag cluster structure is associated with the largest eigenvalues of the $\tilde{A}$, the permutation test statistic is the largest eigenvalue of $\tilde{A}$
- reject the null hypothesis with p-value $p < 0.005$, and conclude that there is significant temporal structure in US equity markets.

Data-driven clustering with known lead-lag mechanisms

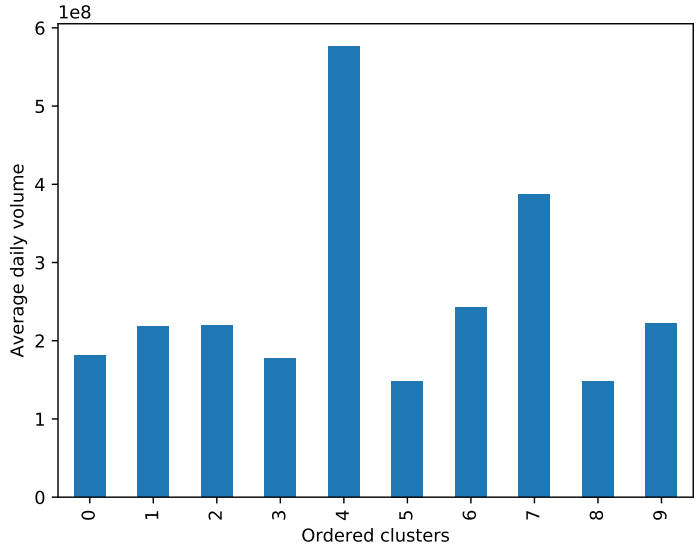
Recovered clusterings **cannot** be explained by the three previously hypothesized mechanisms in the empirical finance lead-lag literature

1. Sector membership induces clustered lead-lag effects
   - Biely and Thurner [2008] fin find associations between sector membership and lead-lag structure on the high-frequency scale of returns
2. Equities with higher trading volume are hypothesized to lead lower volume equities
   - disparities in trading volume across equities can lead to non-synchronous trading lead-lag effects
   - clustering structure may be induced by ordering equities based on quantiles of average trading volume
3. Larger capitalization equities are hypothesized to lead lower capitalization equities; this market cap mechanism can produce lead-lag effects partly via non-trading effects and partly via other channels
   - large stocks may lead small stocks via volatility spillovers.
   - clustering structure may be induced by ordering equities based on quantiles of market capitalization.

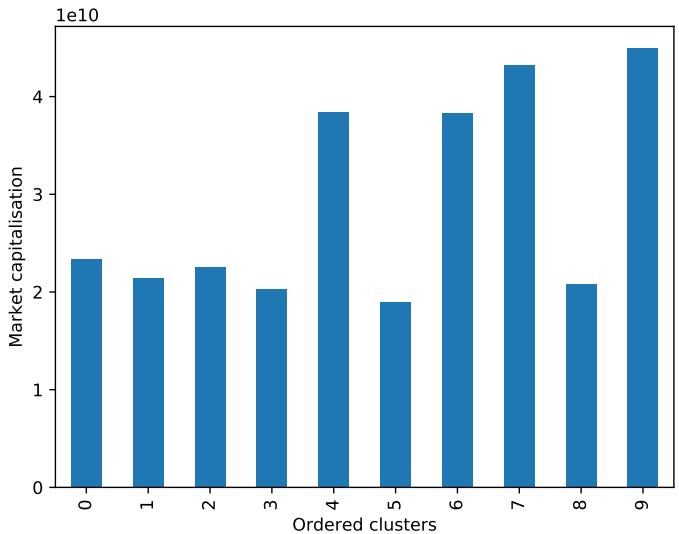
Data-driven clustering with known lead-lag mechanisms



(a) Hist. of Herm-RW and SIC clustering meta-flow edge weights.



(b) Average daily dollar volume by Hermitian RW cluster.



(c) Average market capitalisation by Hermitian RW cluster.

Time variation in the recovered clusterings

Recompute the clustering year-by-year using only data from the retrospective year to do so.

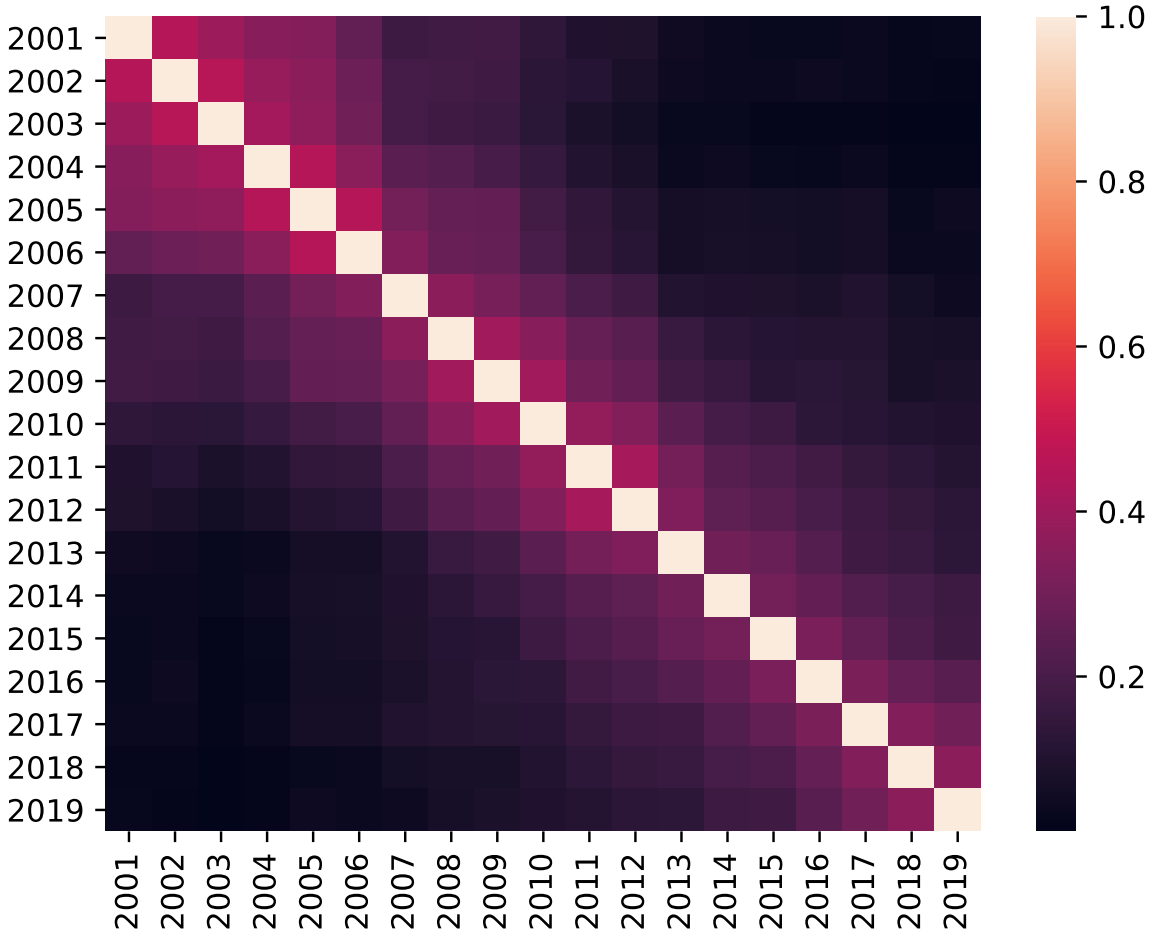


Figure: Adjusted Rand index between clusters computed on yearly snapshots of data

The relatively low ARI values between pairs of clusters indicates some –albeit low– persistence in year-to-year lead-lag structure.

MetaCluster lead-lag portfolios: strategy & cumulative P&L

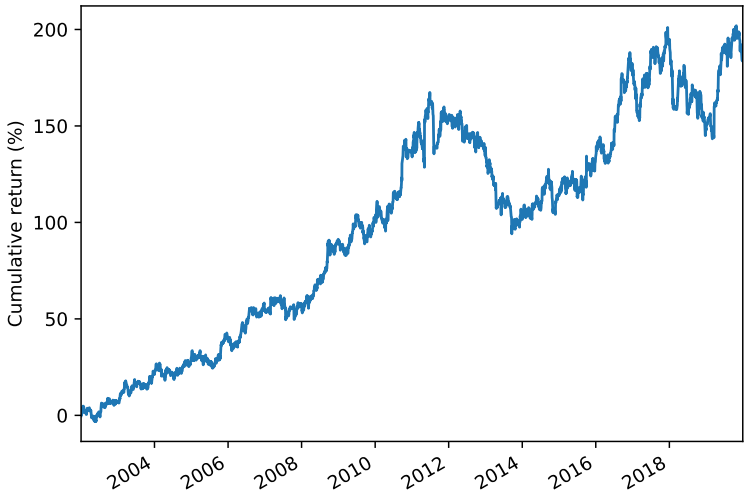


Figure: Cumulative return for the financial forecasting signal.

- MetaCluster: trading signal forecasts lagging cluster returns using smoothed leading cluster returns (update clusters every 2 months); consider only top 10% strongest lead-lag relationships between clusters

MetaCluster lead-lag portfolios: strategy & cumulative P&L

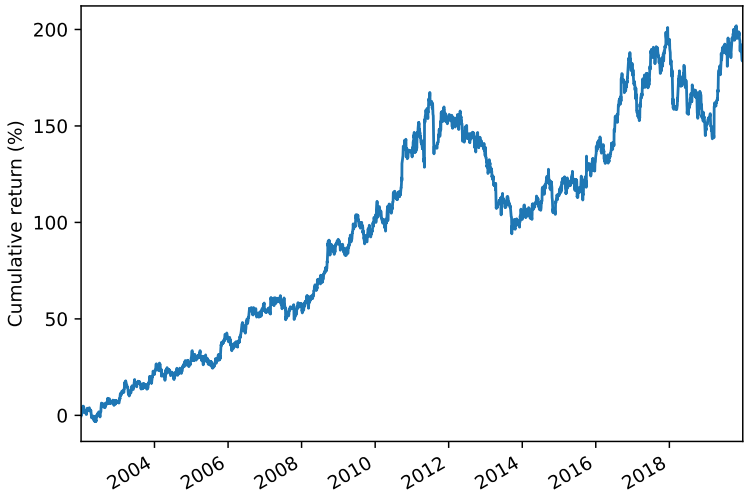


Figure: Cumulative return for the financial forecasting signal.

- MetaCluster: trading signal forecasts lagging cluster returns using smoothed leading cluster returns (update clusters every 2 months); consider only top 10% strongest lead-lag relationships between clusters
- annualized Sharpe Ratio of 0.62 (the S&P500 market return has SR=0.4 on the same period
- low correlation (4%) with the market return
- mean pnl return of 2.4 bpts/day
- decay in the performance of the signal after 2012;
- can be compared with the reduction in clustering persistence post 2012
- in line with Curme et al. (2015) - the information efficiency of the market appears to increase in 2012 relative to earlier years

Data-driven lead-lag detection pipeline

- we propose a data-driven pipeline aiming to capture the fast-changing dynamic inherent in lead-lag relationships
- our methods are capable of detecting and verifying lead-lag relationships effectively and deliver superior portfolio performances

Outline:

1. measure pairwise lead-lag relationships
2. extrapolate pairwise relationships to higher-order relationships
   - (A) MetaCluster: build cluster-driven portfolios that lead-lag each other, or
   - (B) GlobalRank: extract ranking-based ”global" leaders and laggers/followers, (Cartea, C, Jin 2023)
   - (C) ClusterRank: combine clustering & ranking for lead-lag extraction
3. construct a portfolio to test/evaluate/exploit the uncovered lead-lag relationship.

GlobalRank Lead-lag Portfolios

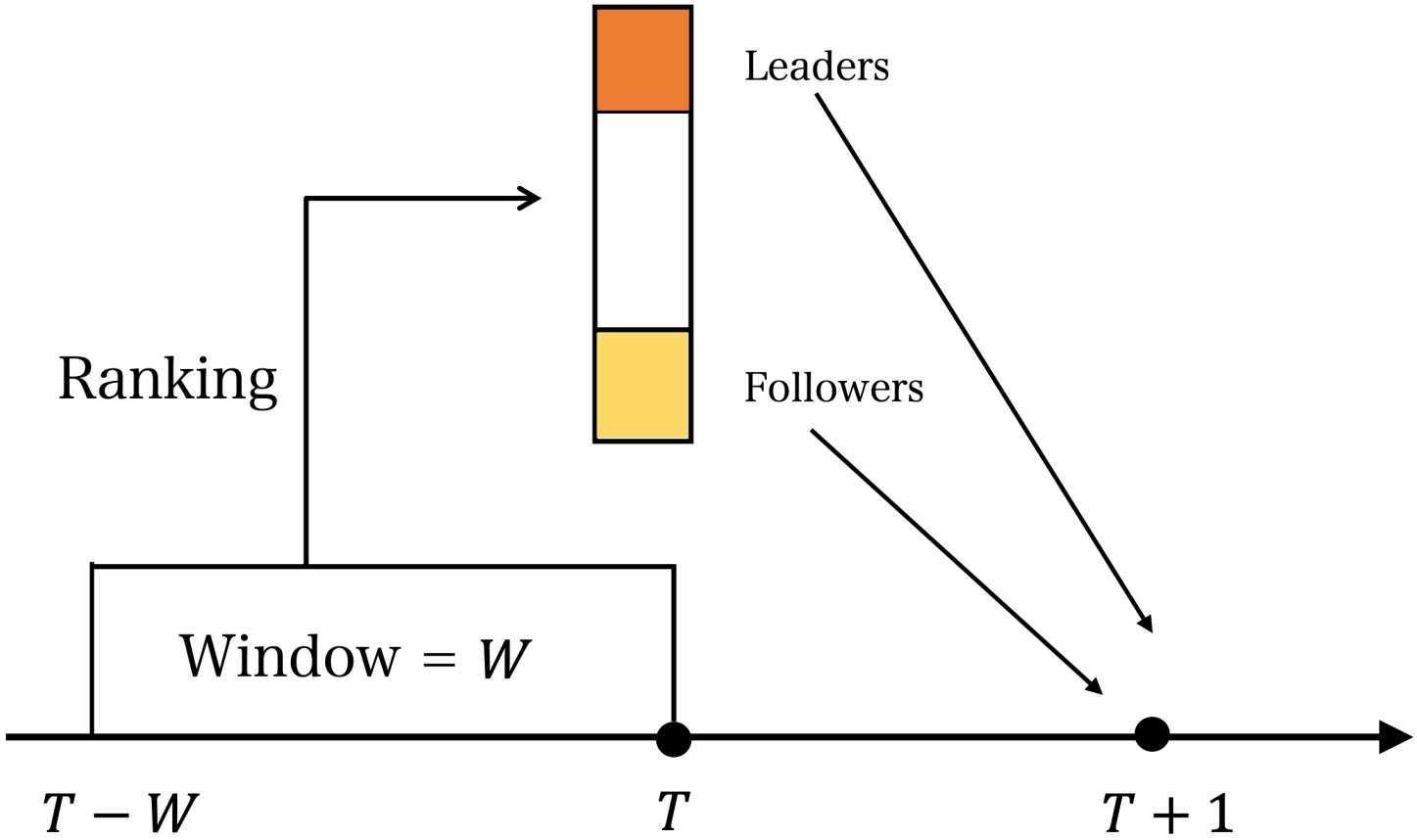


Figure: Illustration of the global ranking pipeline.

- use past 1-day CLCL return of the leaders as indicator of the future return of followers

GlobalRank Lead-lag Portfolios

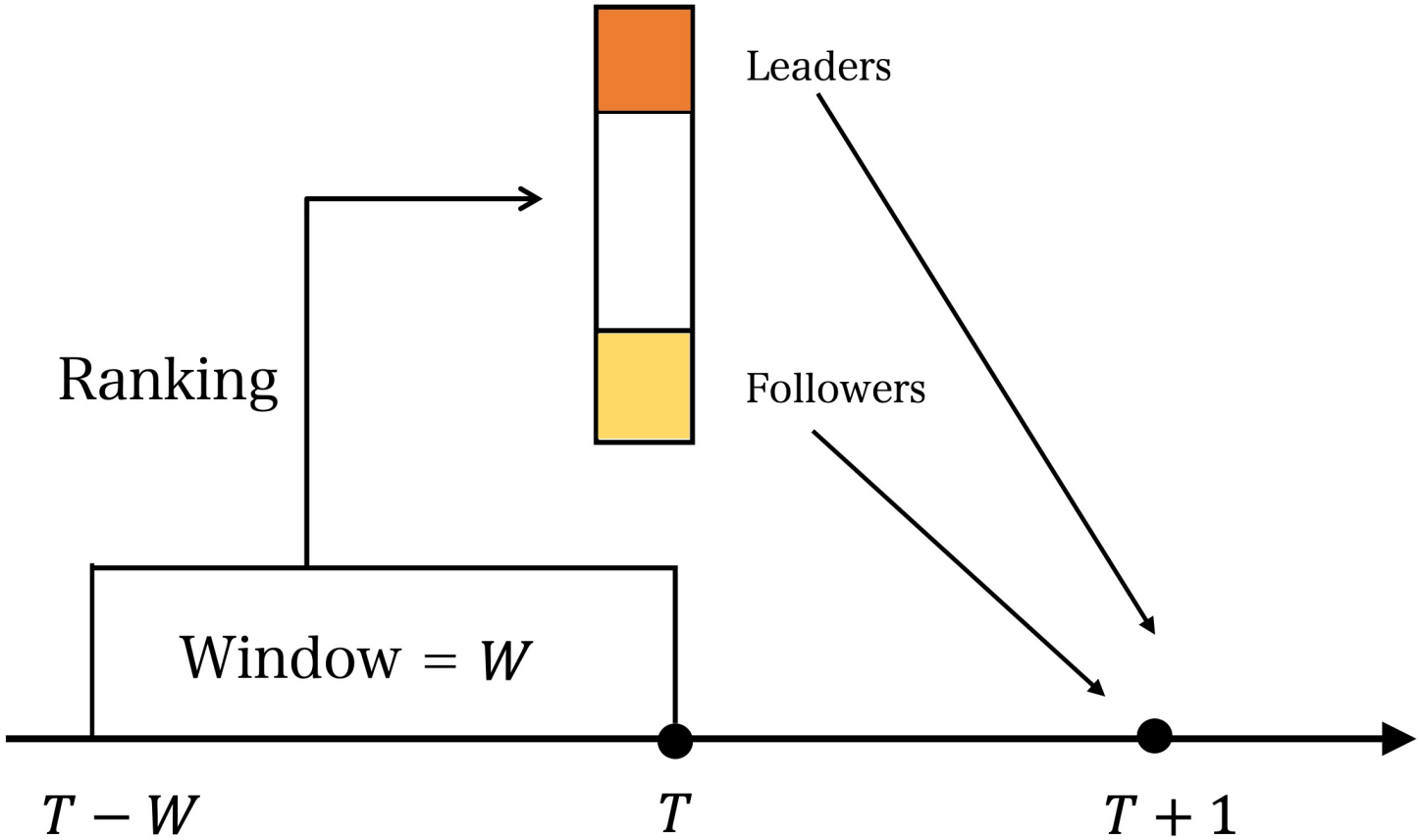


Figure: Illustration of the global ranking pipeline.

- use past 1-day CLCL return of the leaders as indicator of the future return of followers
- to keep the portfolio dollar neutral, trade SPY on the opposite direction of the signal

Fitting previous works into our framework

- interpret the ”heuristic" models as to have derived their scoring mechanism from financial intuition, and to have ranked the assets by the column average of the pairwise lead-lag matrix.

- Lo and MacKinlay (1990):
  - lead-lag score = difference between market caps of asset *i* and asset *j*
  - ranking = column average (i.e. net difference in market cap compared to all other assets)

- Chordia and Swaminathan (2000):
  - lead-lag score = difference between turnover ratios of asset *i* and asset *j*
  - ranking = column average (i.e. net difference in turnover ratio compared to all other assets)

Data-driven lead-lag detection pipeline

- we propose a data-driven pipeline aiming to capture the fast-changing dynamic inherent in lead-lag relationships
- our methods are capable of detecting and verifying lead-lag relationships effectively and deliver superior portfolio performances

Outline:

1. measure pairwise lead-lag relationships
2. extrapolate pairwise relationships to higher-order relationships
   - (A) MetaCluster: build cluster-driven portfolios that lead-lag each other, or
   - (B) GlobalRank: extract ranking-based ”global" leaders and laggers/followers, or
   - (C) ClusterRank: combine clustering & ranking for lead-lag extraction, (Cartea, C, Jin 2023)
3. construct a portfolio to test/evaluate/exploit the uncovered lead-lag relationship.

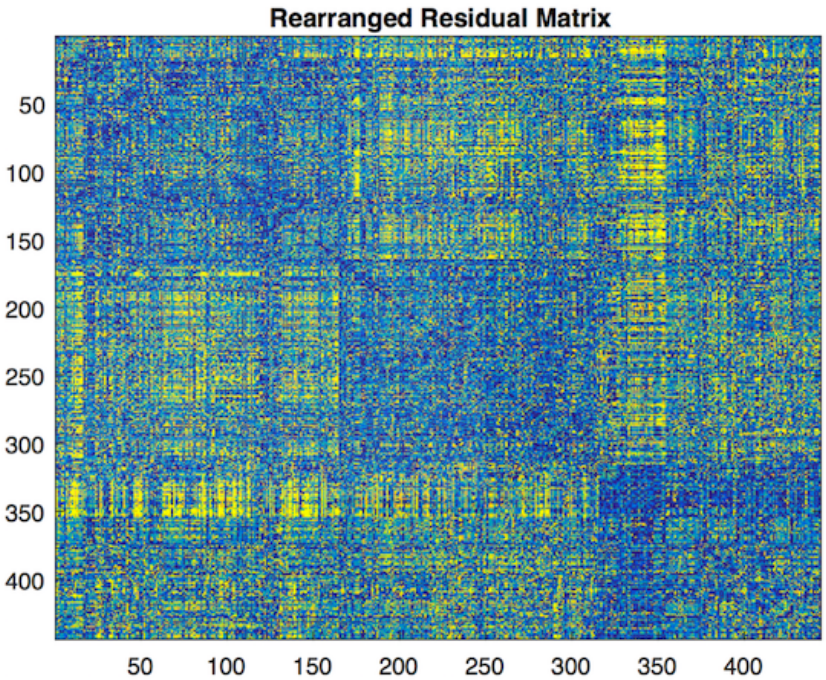
Extraction of partial rankings

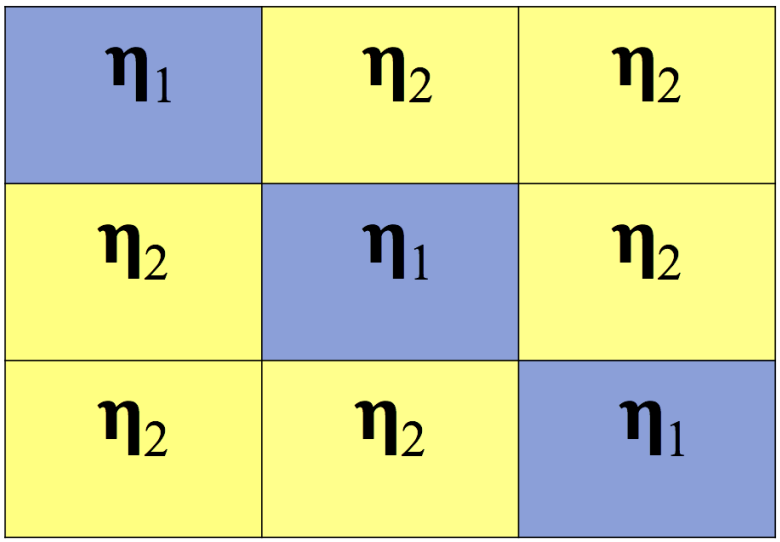
Goal: extract partial rankings - detect clusters within which the lead-lag relationships are more consistent w.r.t an underlying ordering.

Extraction of partial rankings

Goal: extract partial rankings - detect clusters within which the lead-lag relationships are more consistent w.r.t an underlying ordering.

Modularity clustering of the residual matrix $R = |C - \hat{C}|$; S&P 500





Noise levels $\eta_1 < \eta_2$

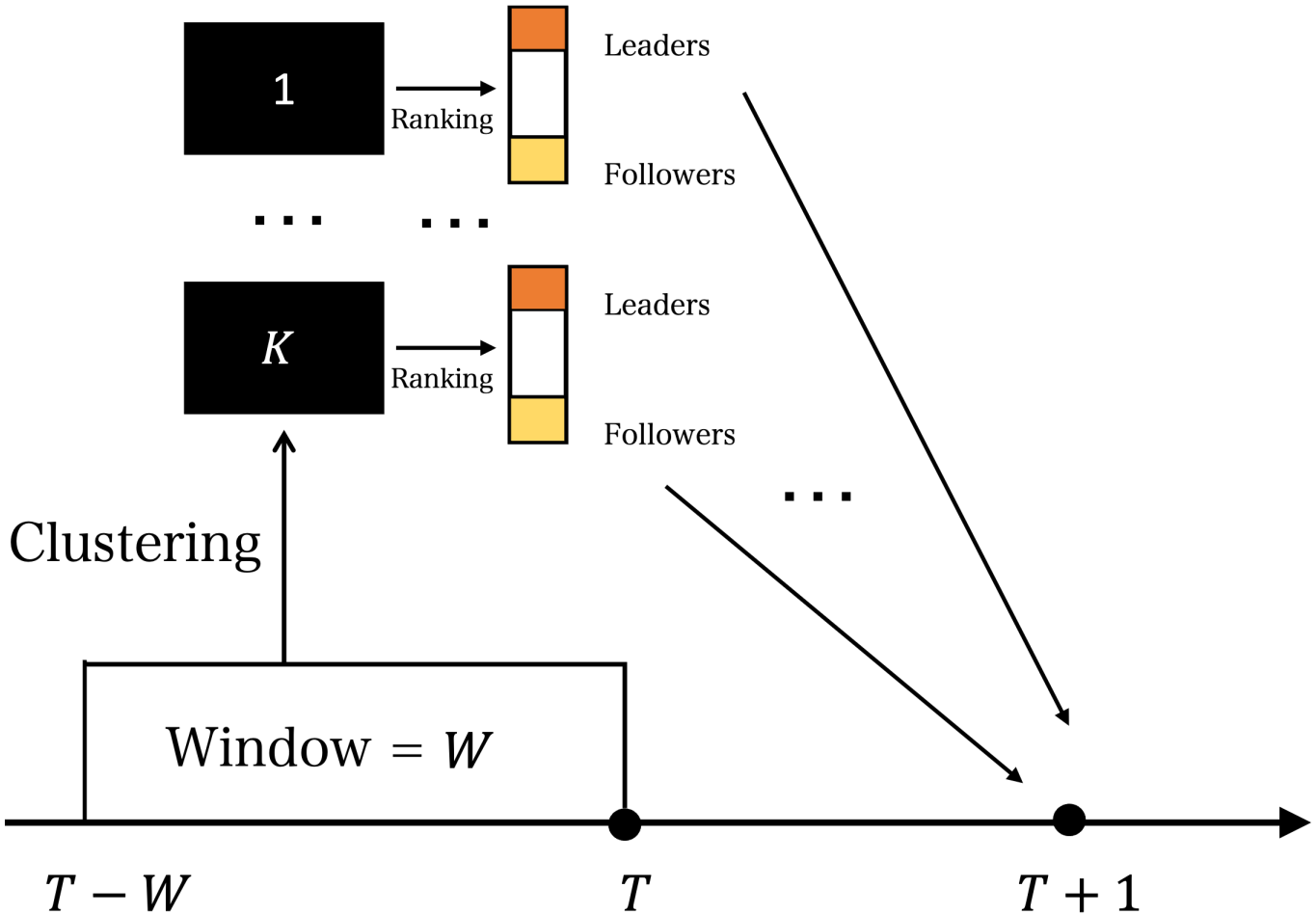
- Stochastic block model for ranking

$$C_{ij} = r_i - r_j + \text{noise}$$

heterogeneous noise (easier to solve (*rank*) within each cluster)

Cluster-Ranking Lead-lag Portfolios

- Hou (2007) and others have studied intra-industry lead-lag relationships



Clustering algorithms on comparison matrices:

- Spectral clustering
- Hermitian clustering

Data set description

- prices and share information data comes from the Center of Research in Security Prices (CRSP) daily prices database.
- sample period from July 1963 to December 2022
- to be consistent with standard literature practices, we include NYSE, Amex, NASDAQ stocks
- for realistic trading simulation results, at each trading day, we only include stocks in the top 25% by market capitalization (defined as stock end-of-day price $\times$ shares outstanding)
- industry classification information is created by linking each firm to a single, non-overlapping Fama-French 12 industry by its SIC code.
  - industries: nondurables (1), durables (2), manufacturing (3), energy (4), chemicals (5), business equipment(6), telecommunications (7), utilities (8), shops (9), healthcare (10), finance (11), and other(12).
- altogether a universe of about 550 assets on each day, on average across time

Empirical results: GlobalRank vs ClusterRank

Table: Performance of various lead-lag portfolios

Panel A: GlobalRank Lead-Lag Portfolios					
	Compound Return (%)	Return (bps/day)	Volatility (%)	Sharpe Ratio	Max Drawdown (%)
Max Cross-Cor	19.69	7.14	0.47	2.37	16.90
Avg Cross-Cor	27.97	9.79	0.70	2.21	28.64
Signature	24.87	8.82	0.59	2.38	24.67
Market Cap	6.15	2.37	0.50	0.75	63.40
Turnover	7.60	2.91	0.34	1.37	13.62
Panel B: ClusterRank Lead-lag Portfolios					
	Compound Return (%)	Return (bps/day)	Volatility (%)	Sharpe Ratio	Max Drawdown (%)
Max Cross-Cor	14.52	5.39	0.38	2.23	18.21
Avg Cross-Cor	19.01	6.91	0.49	2.23	23.49
Signature	17.50	6.40	0.42	2.43	12.28
Industry	15.75	5.81	0.43	2.18	42.29

Param. Specs in GlobalRank & ClusterRank Lead-lag Portfolios I

Table: Performances of Different Lead-lag Portfolios - Alternative Hyperparameters

Panel A: GlobalRank Lead-Lag Portfolios - 40% Leaders and Followers

	Compound Return (%)	Return (bps/day)	Volatility (%)	Sharpe Ratio	Max Drawdown (%)
Max Cross-Cor	18.36	6.70	0.44	2.43	18.9
Avg Cross-Cor	21.37	7.69	0.48	2.54	18.2
Signature	24.26	8.62	0.51	2.70	17.8
Market Cap	0.08	0.03	0.46	0.01	91.75
Turnover	8.90	3.38	0.35	1.52	13.93

Panel B: ClusterRank Lead-lag Portfolios (Hermitian Clustering) - 20% Leader and Lagger each

	Compound Return (%)	Return (bps/day)	Volatility (%)	Sharpe Ratio	Max Drawdown (%)
Max Cross-Cor	15.73	5.80	0.38	2.38	12.46
Avg Cross-Cor	21.74	7.81	1.18	1.05	38.20
Signature	24.31	8.64	0.39	3.43	9.19

Panel C: ClusterRank Lead-lag Portfolios (Spectral Clustering) - 40% Leader and Lagger each

	Compound Return (%)	Return (bps/day)	Volatility (%)	Sharpe Ratio	Max Drawdown (%)
Max Cross-Cor	15.06	5.57	0.35	2.42	11.75
Avg Cross-Cor	17.23	6.31	0.40	2.52	20.39
Signature	19.65	7.12	0.38	2.99	11.51
Industry	6.64	2.55	0.38	1.06	26.9

Param. Specs in GlobalRank & ClusterRank Lead-lag Portfolios II

Table: Performances of Different Lead-lag Portfolios - Alternative Hyperparameters

Panel D: ClusterRank Lead-lag Portfolios (Hermitian Clustering) - 40% Leader and Lagger					
	Compound Return (%)	Return (bps/day)	Volatility (%)	Sharpe Ratio	Max Drawdown (%)
Max Cross-Cor	16.08	5.92	0.37	2.51	14.77
Avg Cross-Cor	20.98	7.56	1.14	1.05	39.49
Signature	22.70	8.12	0.37	3.45	10.07
Panel E: GlobalRank Lead-lag Portfolios - 20% Leader and Lagger, $W = 30$ Day Look-back Window					
	Compound Return (%)	Return (bps/day)	Volatility (%)	Sharpe Ratio	Max Drawdown (%)
Max Cross-Cor	29.60	10.29	0.74	2.21	27.48
Avg Cross-Cor	22.72	8.13	0.51	2.19	16.91
Signature	23.64	8.42	0.59	2.61	16.72

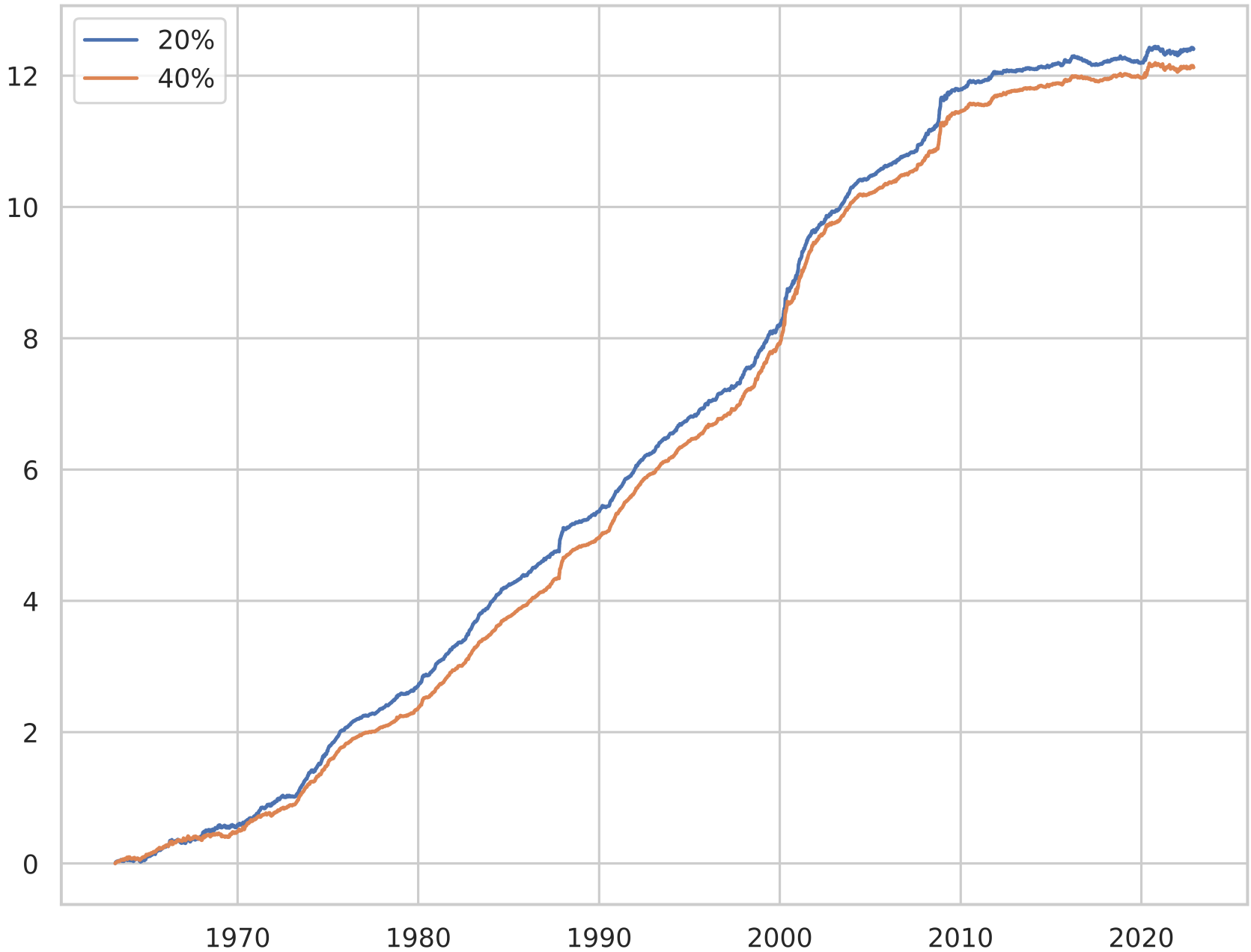
Param. Specs in GlobalRank & ClusterRank Lead-lag Portfolios II

Table: Performances of Different Lead-lag Portfolios - Alternative Hyperparameters

	Compound Return (%)	Return (bps/day)	Volatility (%)	Sharpe Ratio	Max Drawdown (%)
Panel D: ClusterRank Lead-lag Portfolios (Hermitian Clustering) - 40% Leader and Lagger					
Max Cross-Cor	16.08	5.92	0.37	2.51	14.77
Avg Cross-Cor	20.98	7.56	1.14	1.05	39.49
Signature	22.70	8.12	0.37	3.45	10.07
Panel E: GlobalRank Lead-lag Portfolios - 20% Leader and Lagger, $W = 30$ Day Look-back Window					
	Compound Return (%)	Return (bps/day)	Volatility (%)	Sharpe Ratio	Max Drawdown (%)
Max Cross-Cor	29.60	10.29	0.74	2.21	27.48
Avg Cross-Cor	22.72	8.13	0.51	2.19	16.91
Signature	23.64	8.42	0.59	2.61	16.72

Results are Robust to Hyper Parameters

Cumsum PnL - Hermitian Clustering 1970-2022



Turnovers analysis

- for a portfolio that trades $w_1^t, ..., w_n^t$ dollar notional for stocks $s_1, ..., s_n$ at time t, the turnover of the portfolio at time t is defined as

$$\text{Turnover}^t = \frac{\sum_{i=1}^{n} |w_i^t - w_i^{t-1}|}{\sum_{i=1}^{n} |w_i^{t-1}|},$$

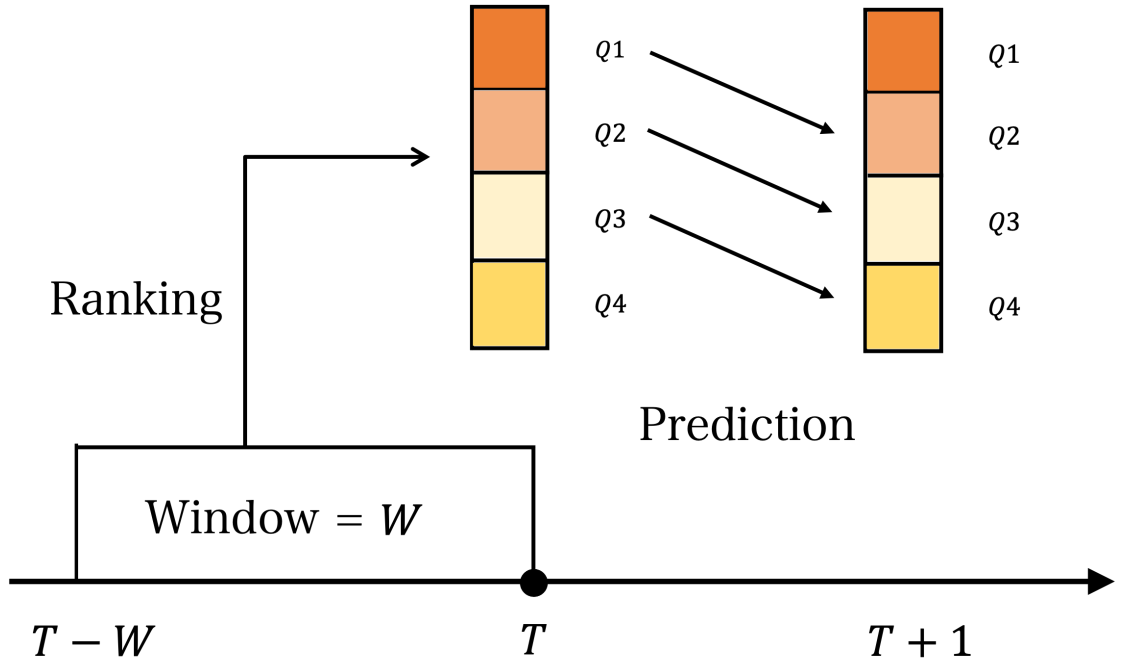
where $|w_i^t - w_i^{t-1}|$ is the change in dollar value held by the portfolio on asset s_i over the re-balance period.

Table: Turnover ratio of various portfolios

	Avg turnover ratio (%)	Avg change in portfolio composition(%)	Proportion of sign flips(%)
Max Cross-Cor	102.5	22.6	44.3
Avg Cross-Cor	104.5	26.7	44.1
Signature	111.3	23.6	50.4
Market Cap	95.4	1.21	44.2
Turnover	94.4	1.22	43.3

Waterfall GlobalRank lead-lag relationships

- what about the middle 60% of stocks not included in the analysis?

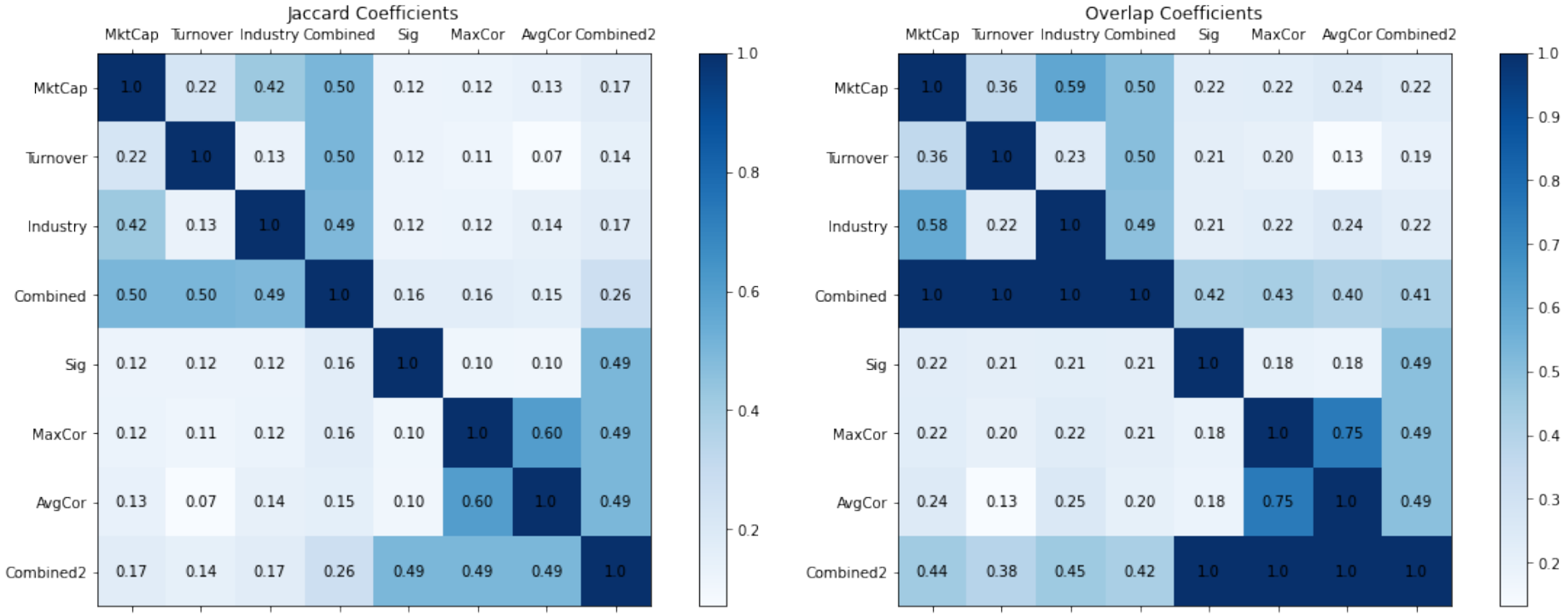


- intermediate lead-lag relationships exist
- leads to larger traded universe

Table: Waterfall Lead-lag Relationships

	Compound Return (%)	Return (bps/day)	Volatility (%)	Sharpe Ratio	Max Drawdown (%)
Max Cross-Cor	16.1	5.91	0.40	2.37	15.8
Avg Cross-Cor	14.8	5.48	0.35	2.52	14.1
Signature	20.4	7.36	0.40	2.89	14.7

Similarity across portfolio types



(a) Jaccard coefficients

(b) Overlap coefficients

Figure: (a) Diagram for the Jaccard coefficients among different lead-lag portfolios; a higher Jaccard coefficient represents higher similarity. (b) Diagram for the overlap coefficients among different lead-lag portfolios; a higher overlap coefficient represents higher similarity. Covered period: 2019-2022.

Clusters composition

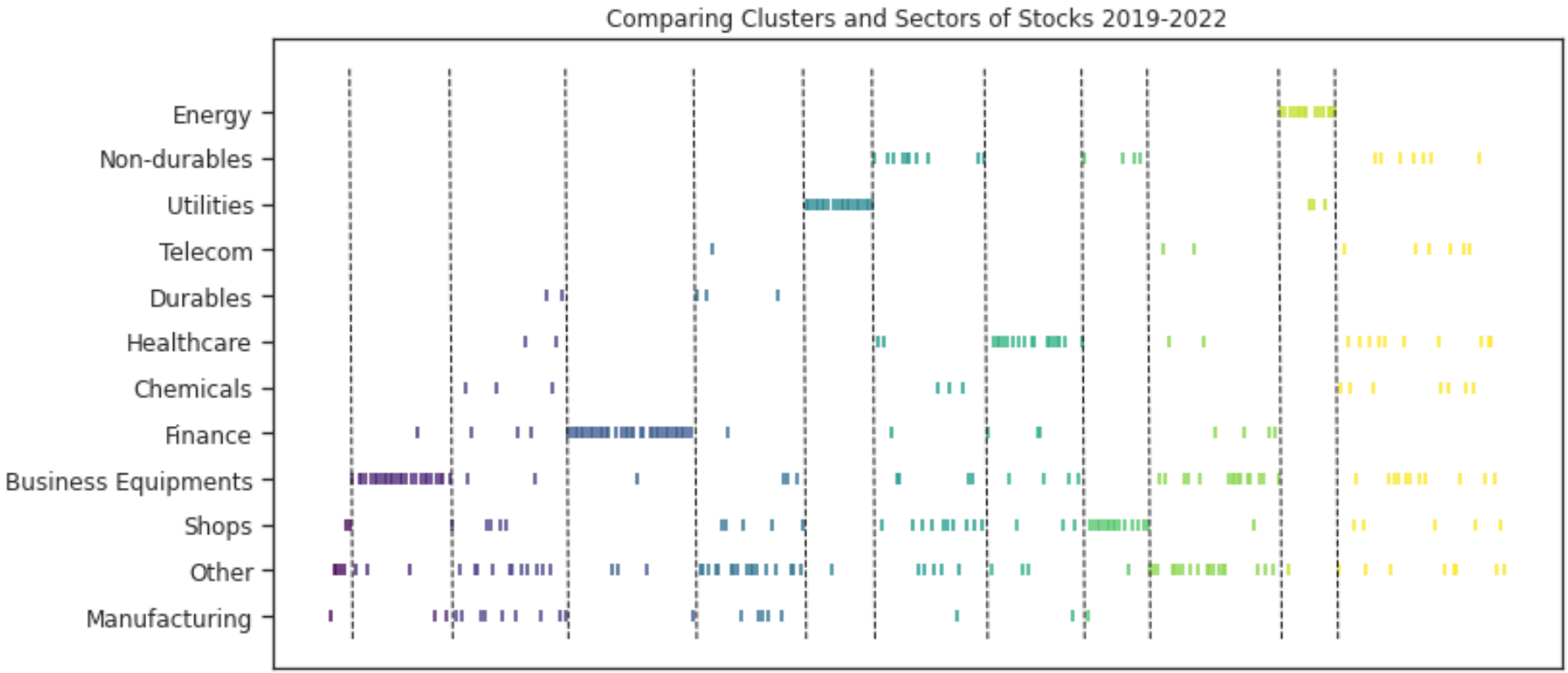
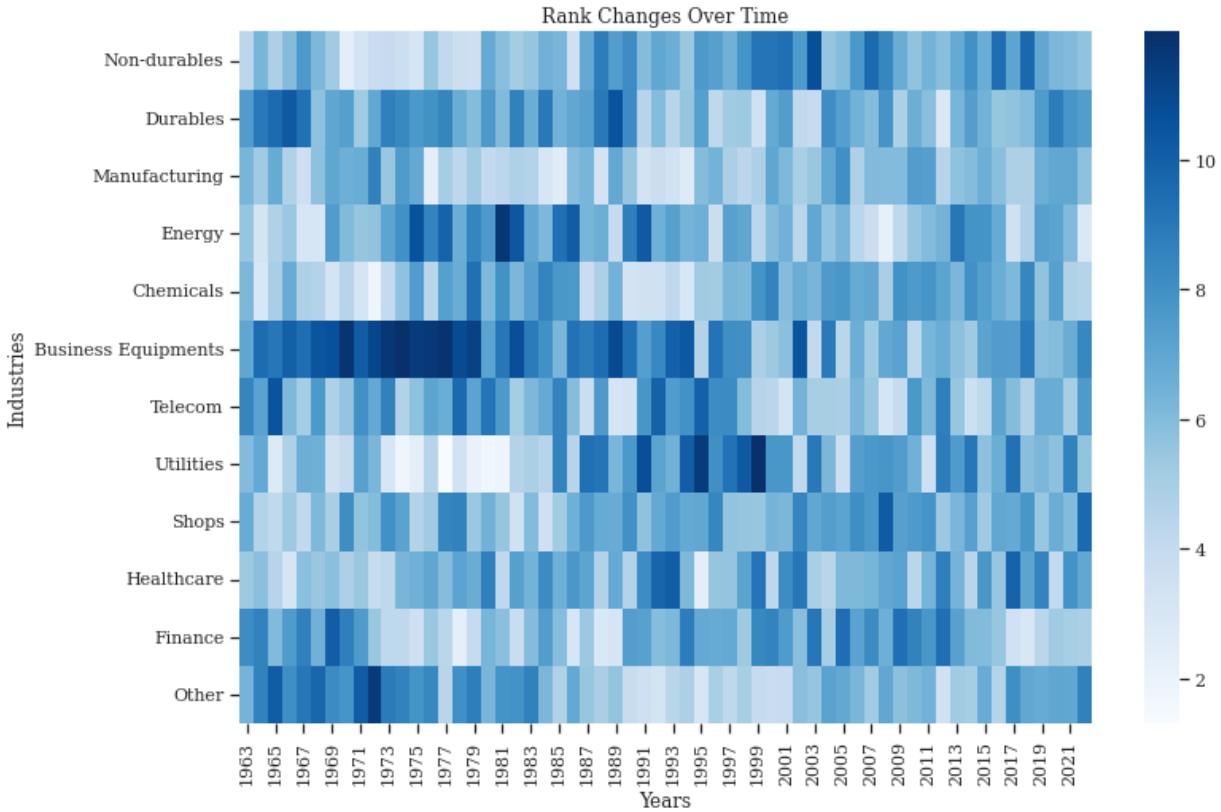


Figure: A comparison between the clusters using the Signature matrices from 1 January 2019 to 31 December 2022. The area between black vertical dashes represents each cluster formed with Hermitian clustering. There are 376 stocks that are traded every day in this time period.

Evolution across time of sector ranks



Manufacturing	0.476
Chemicals	0.485
Energy	0.494
Telecom	0.495
Healthcare	0.496
Utilities	0.498
Non-durables	0.499
Shops	0.501
Finance	0.503
Other	0.504
Durables	0.511
Business Equipments	0.554

Figure: Left: rank evolution of sector averages. Right: average rank across time (sorted from most leading to most lagging sectors).

Revisiting the slow information diffusion hypothesis

- if the hypothesis holds, lead-lag relationship should slowly disappear as we reduce the time frequency by which we detect the relationship/ rebalance the portfolio.

Revisiting the slow information diffusion hypothesis

- if the hypothesis holds, lead-lag relationship should slowly disappear as we reduce the time frequency by which we detect the relationship/ rebalance the portfolio.

- explore a multiscale approach, and consider daily, bi-daily, weekly, bi-weekly, tri-weekly, and monthly portfolios.

Revisiting the slow information diffusion hypothesis

- if the hypothesis holds, lead-lag relationship should slowly disappear as we reduce the time frequency by which we detect the relationship/ rebalance the portfolio.
- explore a multiscale approach, and consider daily, bi-daily, weekly, bi-weekly, tri-weekly, and monthly portfolios.
- eg. for monthly, consider the previous-month cumulative return of the leaders, and use that to predict return of the followers over the next month

Lead-lag relationships at multiple frequencies

Table: Performance of lead-lag portfolios at various frequencies (return is per period).

Panel A: Signature Portfolio for various frequencies

	Compound Return(%)	Return (bps/per)	Volatility(%)	Sharpe Ratio	Max Drawdown(%)
Daily	24.87	8.82	0.59	2.37	24.67
Bi-Daily	11.02	8.30	0.76	1.23	24.17
Weekly	12.33	23.09	1.34	1.22	26.65
Bi-Weekly	8.37	31.94	1.69	0.95	19.59
Tri-Weekly	7.30	42.9	2.01	0.87	24.56
Monthly	7.81	65.8	2.56	0.86	22.50

Panel B: MaxCor Portfolio for various frequencies

	Compound Return(%)	Return (bps/per)	Volatility(%)	Sharpe Ratio	Max Drawdown(%)
Daily	19.69	7.14	0.48	2.36	16.90
Bi-Daily	13.36	9.96	0.66	1.70	32.64
Weekly	13.32	24.84	1.09	1.62	28.48
Bi-Weekly	9.51	36.12	1.47	1.24	23.16
Tri-Weekly	8.86	50.64	1.80	1.15	16.95
Monthly	6.91	61.20	2.35	0.86	21.5

Panel C: AvgCor Portfolio for various frequencies

	Compound Return(%)	Return (bps/per)	Volatility(%)	Sharpe Ratio	Max Drawdown(%)
Daily	27.97	9.79	0.70	2.21	28.64
Bi-Daily	16.46	12.10	0.96	1.41	52.67
Weekly	16.41	30.19	1.56	1.38	37.26
Bi-Weekly	12.63	47.31	1.99	1.19	33.79
Tri-Weekly	11.65	65.80	2.39	1.13	24.91
Monthly	9.76	85.36	3.02	0.94	19.52

A multi-frequency perspective

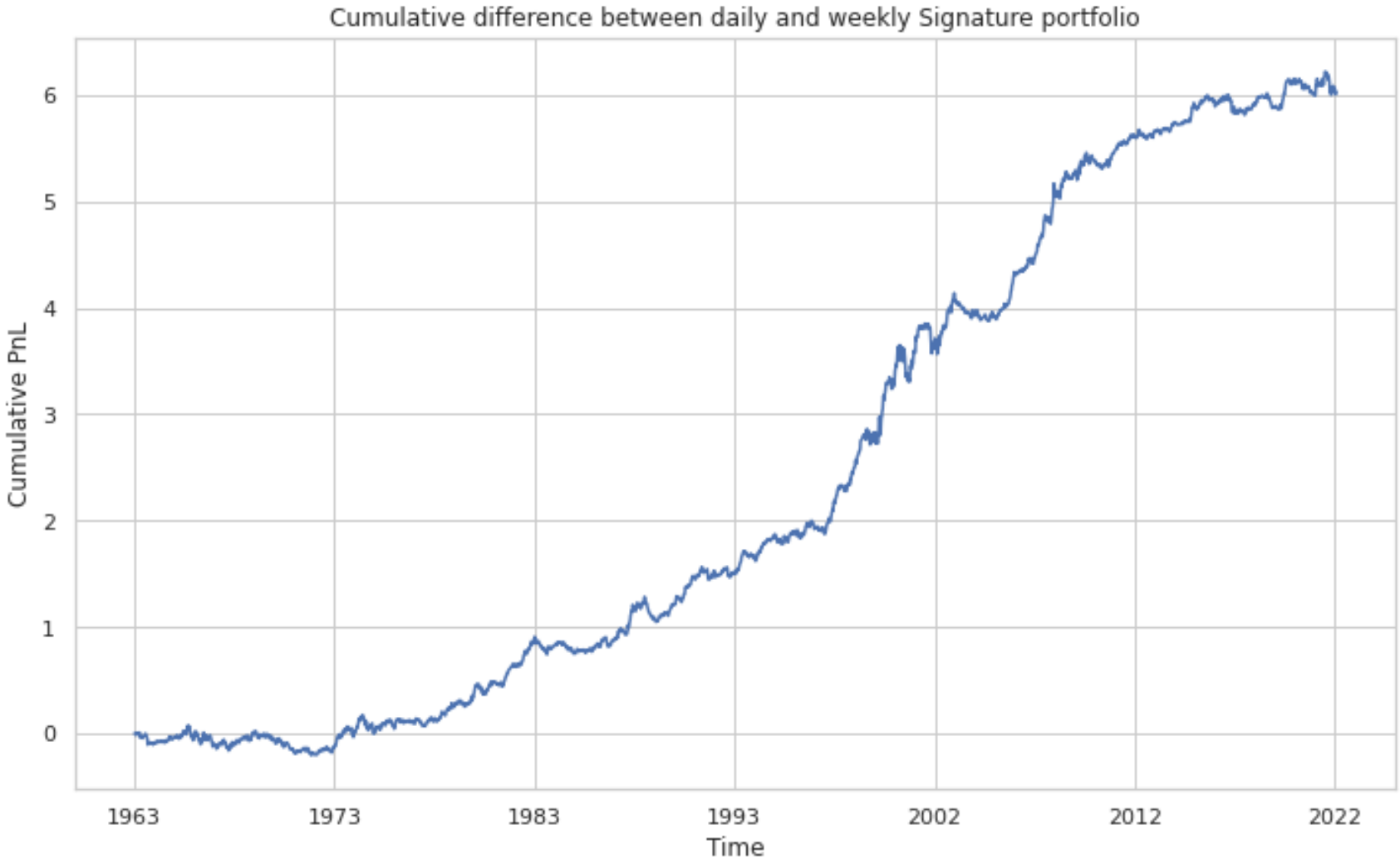


Figure: Cumulative sum of the difference between the daily lead-lag Signature portfolio and the weekly lead-lag Signature portfolio

Historical progression of lead-lag relationships

Table: Dates when daily lead-lag portfolios first out-perform lower frequency lead-lag portfolios by a certain margin.

Panel A: First occurrence of a cumulative excess return of 50% btw daily and lower-frequency portfolios

	Raw	Bi-Daily	Weekly	Bi-Weekly	Tri-Weekly	Monthly
Signature	1968.07	1973.04	1981.09	1980.09	1978.10	1971.02
Max Cross-Corr	1968.03	1974.07	1996.02	1982.07	1975.08	1970.08
Avg Cross-Corr	1968.02	1974.01	1995.06	1985.02	1982.08	1973.12

Panel B: First occurrence of a cumulative excess return of 100% btw daily and lower-frequency portfolios

	Raw	Bi-Daily	Weekly	Bi-Weekly	Tri-Weekly	Monthly
Signature	1973.04	1974.09	1987.10	1981.11	1981.03	1974.12
Max Cross-Corr	1971.04	1990.08	1999.08	1985.09	1983.02	1974.07
Avg Cross-Corr	1970.06	1982.12	1996.02	1987.10	1984.07	1981.08

Historical progression of lead-lag relationships

Table: Dates when daily lead-lag portfolios first out-perform lower frequency lead-lag portfolios by a certain margin.

Panel A: First occurrence of a cumulative excess return of 50% btw daily and lower-frequency portfolios

	Raw	Bi-Daily	Weekly	Bi-Weekly	Tri-Weekly	Monthly
Signature	1968.07	1973.04	1981.09	1980.09	1978.10	1971.02
Max Cross-Corr	1968.03	1974.07	1996.02	1982.07	1975.08	1970.08
Avg Cross-Corr	1968.02	1974.01	1995.06	1985.02	1982.08	1973.12

Panel B: First occurrence of a cumulative excess return of 100% btw daily and lower-frequency portfolios

	Raw	Bi-Daily	Weekly	Bi-Weekly	Tri-Weekly	Monthly
Signature	1973.04	1974.09	1987.10	1981.11	1981.03	1974.12
Max Cross-Corr	1971.04	1990.08	1999.08	1985.09	1983.02	1974.07
Avg Cross-Corr	1970.06	1982.12	1996.02	1987.10	1984.07	1981.08

Potential hypotheses:

- historically, the market got faster and faster - information that used to take weeks or months to arrive at the followers now takes much less time, causing the daily lead-lag portfolio to ultimately outperform the lower frequency ones.

Historical progression of lead-lag relationships

Table: Dates when daily lead-lag portfolios first out-perform lower frequency lead-lag portfolios by a certain margin.

Panel A: First occurrence of a cumulative excess return of 50% btw daily and lower-frequency portfolios

	Raw	Bi-Daily	Weekly	Bi-Weekly	Tri-Weekly	Monthly
Signature	1968.07	1973.04	1981.09	1980.09	1978.10	1971.02
Max Cross-Corr	1968.03	1974.07	1996.02	1982.07	1975.08	1970.08
Avg Cross-Corr	1968.02	1974.01	1995.06	1985.02	1982.08	1973.12

Panel B: First occurrence of a cumulative excess return of 100% btw daily and lower-frequency portfolios

	Raw	Bi-Daily	Weekly	Bi-Weekly	Tri-Weekly	Monthly
Signature	1973.04	1974.09	1987.10	1981.11	1981.03	1974.12
Max Cross-Corr	1971.04	1990.08	1999.08	1985.09	1983.02	1974.07
Avg Cross-Corr	1970.06	1982.12	1996.02	1987.10	1984.07	1981.08

Potential hypotheses:

- historically, the market got faster and faster - information that used to take weeks or months to arrive at the followers now takes much less time, causing the daily lead-lag portfolio to ultimately outperform the lower frequency ones.
  - how do we measure the speed of the market?

Historical progression of lead-lag relationships

Table: Dates when daily lead-lag portfolios first out-perform lower frequency lead-lag portfolios by a certain margin.

Panel A: First occurrence of a cumulative excess return of 50% btw daily and lower-frequency portfolios

	Raw	Bi-Daily	Weekly	Bi-Weekly	Tri-Weekly	Monthly
Signature	1968.07	1973.04	1981.09	1980.09	1978.10	1971.02
Max Cross-Corr	1968.03	1974.07	1996.02	1982.07	1975.08	1970.08
Avg Cross-Corr	1968.02	1974.01	1995.06	1985.02	1982.08	1973.12

Panel B: First occurrence of a cumulative excess return of 100% btw daily and lower-frequency portfolios

	Raw	Bi-Daily	Weekly	Bi-Weekly	Tri-Weekly	Monthly
Signature	1973.04	1974.09	1987.10	1981.11	1981.03	1974.12
Max Cross-Corr	1971.04	1990.08	1999.08	1985.09	1983.02	1974.07
Avg Cross-Corr	1970.06	1982.12	1996.02	1987.10	1984.07	1981.08

Potential hypotheses:

- historically, the market got faster and faster - information that used to take weeks or months to arrive at the followers now takes much less time, causing the daily lead-lag portfolio to ultimately outperform the lower frequency ones.
  - how do we measure the speed of the market?
  - it is difficult to establish causal relationship in finance!

Historical progression of lead-lag relationships

Table: Dates when daily lead-lag portfolios first out-perform lower frequency lead-lag portfolios by a certain margin.

Panel A: First occurrence of a cumulative excess return of 50% btw daily and lower-frequency portfolios

	Raw	Bi-Daily	Weekly	Bi-Weekly	Tri-Weekly	Monthly
Signature	1968.07	1973.04	1981.09	1980.09	1978.10	1971.02
Max Cross-Corr	1968.03	1974.07	1996.02	1982.07	1975.08	1970.08
Avg Cross-Corr	1968.02	1974.01	1995.06	1985.02	1982.08	1973.12

Panel B: First occurrence of a cumulative excess return of 100% btw daily and lower-frequency portfolios

	Raw	Bi-Daily	Weekly	Bi-Weekly	Tri-Weekly	Monthly
Signature	1973.04	1974.09	1987.10	1981.11	1981.03	1974.12
Max Cross-Corr	1971.04	1990.08	1999.08	1985.09	1983.02	1974.07
Avg Cross-Corr	1970.06	1982.12	1996.02	1987.10	1984.07	1981.08

Potential hypotheses:

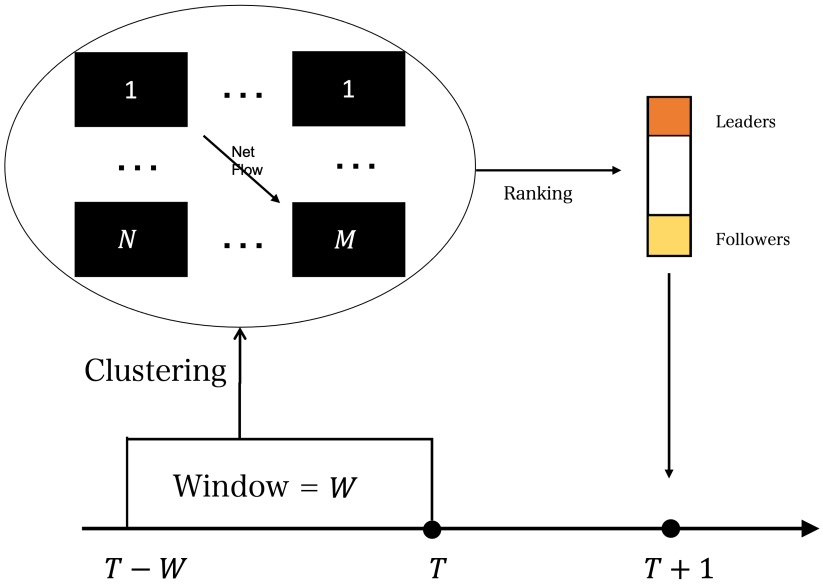
- historically, the market got faster and faster - information that used to take weeks or months to arrive at the followers now takes much less time, causing the daily lead-lag portfolio to ultimately outperform the lower frequency ones.
  - how do we measure the speed of the market?
  - it is difficult to establish causal relationship in finance!
- early days, only few had the capacity to trade daily or even intraday; but if you already can trade bi-daily, there's no reason to not go daily; caused the bi-daily portfolio to be outperformed by the daily one right away.

Data-driven lead-lag detection pipeline

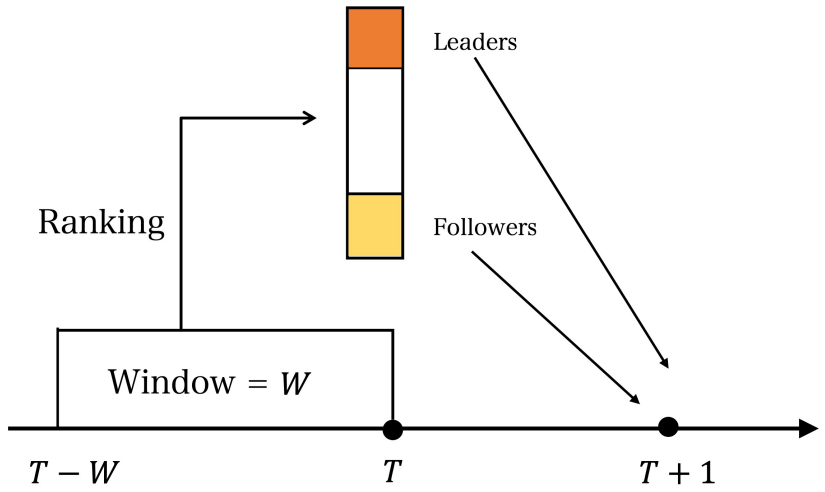
Outline:

1. measure pairwise lead-lag relationships
2. extrapolate pairwise relationships to higher-order relationships
   - (A) MetaCluster: build cluster-driven portfolios that lead-lag each other, or
   - (B) GlobalRank: extract ranking-based ”global" leaders and laggers/followers, or
   - (C) ClusterRank: combine clustering & ranking for lead-lag extraction
3. construct a portfolio to test/evaluate/exploit the uncovered lead-lag relationship.

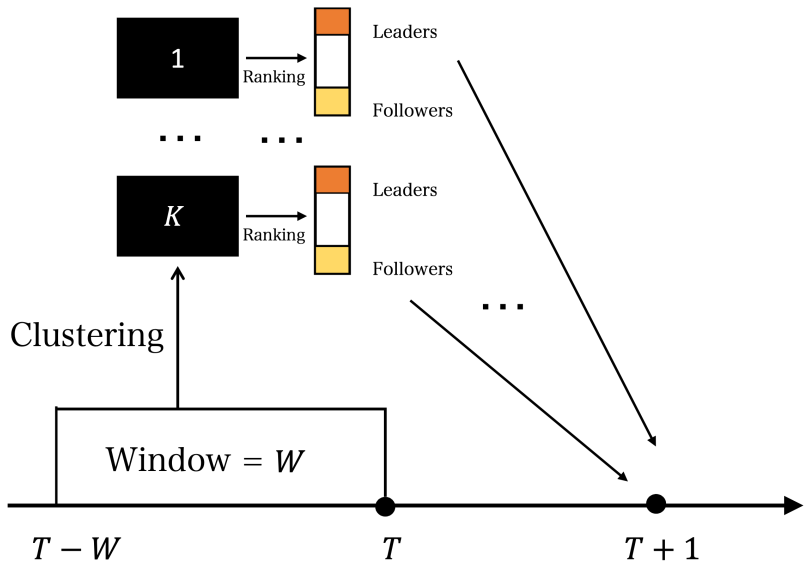
Lead-lag detection frameworks



(a) MetaCluster Lead-lag



(b) GlobalRank Lead-lag



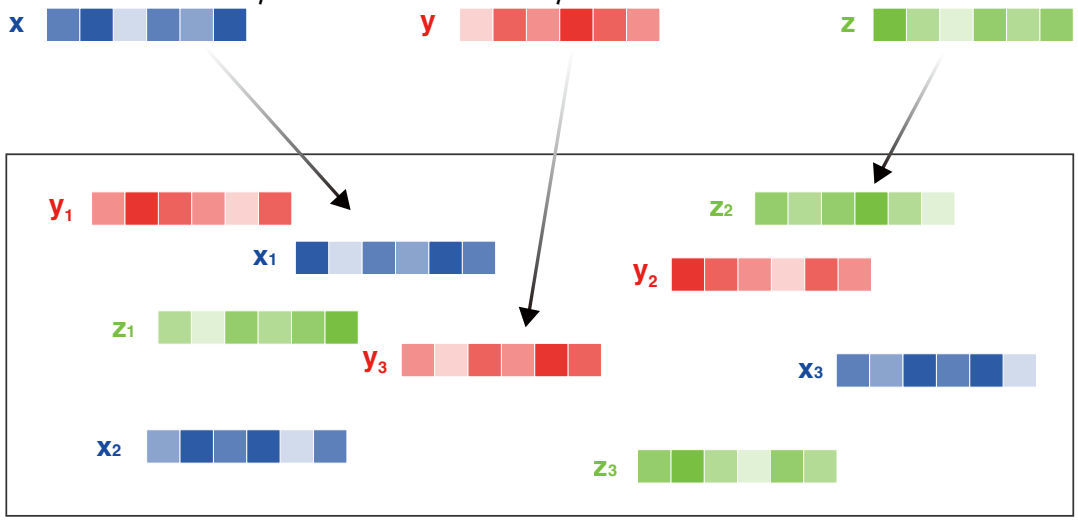
(c) ClusterRank Lead-lag

Multi-reference alignment & multifactor models with lags

- one unknown signal x; samples are shifted noisy versions $\{x_1, \ldots, x_n\}$

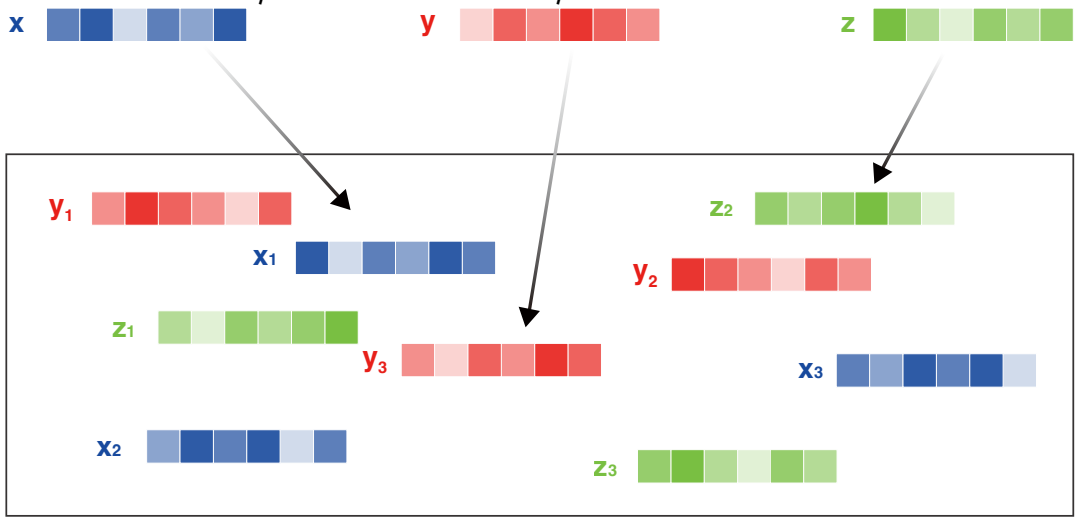
Multi-reference alignment & multifactor models with lags

- one unknown signal x; samples are shifted noisy versions $\{x_1, \ldots, x_n\}$
- multiple unknown signals x,y,z; samples are shifted noisy versions $\{x_1, \ldots, x_{n/3}, y_1, \ldots, y_{n/3}, z_1, \ldots, z_{n/3}\}$



Multi-reference alignment & multifactor models with lags

- one unknown signal x; samples are shifted noisy versions $\{x_1, \ldots, x_n\}$
- multiple unknown signals x,y,z; samples are shifted noisy versions $\{x_1, \ldots, x_{n/3}, y_1, \ldots, y_{n/3}, z_1, \ldots, z_{n/3}\}$



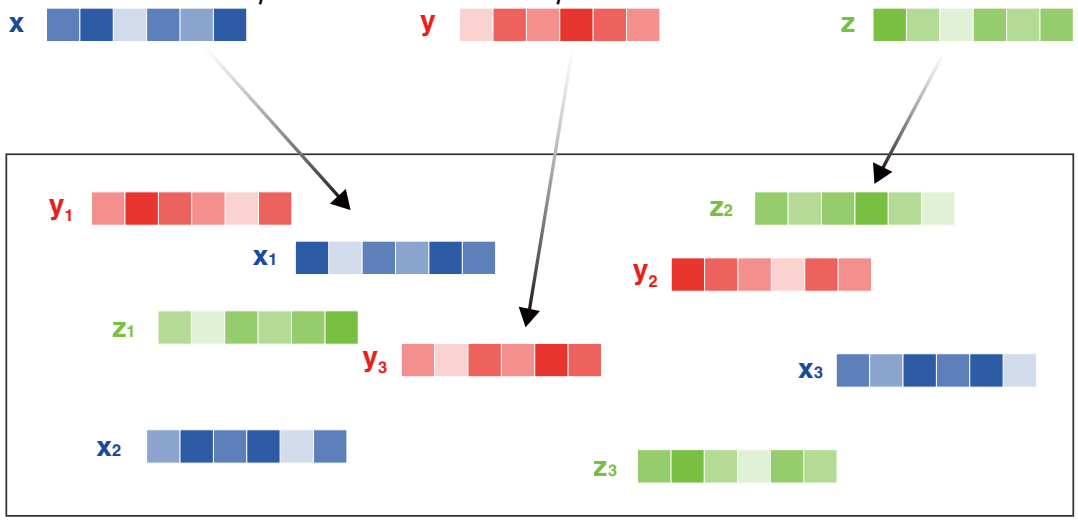
Multifactor models:

- $r_{n\times 1}$: cross-sectional returns of instruments
- $L_{n\times k}$: loadings matrix, with $L_{i,j}$ is the exposure of instrument i to factor j
- $f_{k\times 1}$: the vector of returns of the k factors

$$r = Lf + \epsilon$$

Multi-reference alignment & multifactor models with lags

- one unknown signal x; samples are shifted noisy versions $\{x_1, \ldots, x_n\}$
- multiple unknown signals x,y,z; samples are shifted noisy versions $\{x_1, \ldots, x_{n/3}, y_1, \ldots, y_{n/3}, z_1, \ldots, z_{n/3}\}$



Multifactor models:

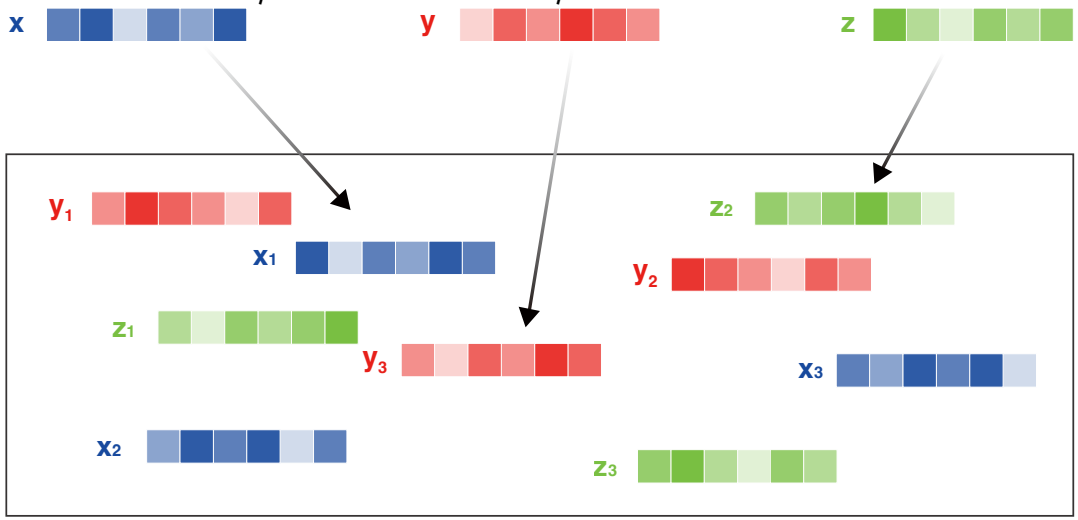
- $r_{n\times 1}$: cross-sectional returns of instruments
- $L_{n\times k}$: loadings matrix, with $L_{i,j}$ is the exposure of instrument i to factor j
- $f_{k\times 1}$: the vector of returns of the k factors

$$r = Lf + \epsilon$$

- time dependent L, time dependent f (AR, ARMA models)

Multi-reference alignment & multifactor models with lags

- one unknown signal x; samples are shifted noisy versions $\{x_1, \ldots, x_n\}$
- multiple unknown signals x,y,z; samples are shifted noisy versions $\{x_1, \ldots, x_{n/3}, y_1, \ldots, y_{n/3}, z_1, \ldots, z_{n/3}\}$



Multifactor models:

- $r_{n\times 1}$: cross-sectional returns of instruments
- $L_{n\times k}$: loadings matrix, with $L_{i,j}$ is the exposure of instrument i to factor j
- $f_{k\times 1}$: the vector of returns of the k factors

$$r = Lf + \epsilon$$

- time dependent L, time dependent f (AR, ARMA models)

Multifactor models with lags:

- $G_{n\times k}$: $G_{i,j}$ lag of instrument i to factor j
- $r_i = L_{i,:} \cdot h(G_{i,:};\ f^t, f^{t-1}, \ldots) + \epsilon_i$

Single membership model with $k = 3$ factors/signals

$$L = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{array}\right] \qquad G = \left[\begin{array}{ccc} 3 & 0 & 0 \\ 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ m & 0 & 0 \\ 2 & 0 & 0 \\ \hline 0 & 5 & 0 \\ 0 & m & 0 \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \\ 0 & 3 & 0 \\ \hline 0 & 0 & 3 \\ 0 & 0 & 2 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 4 \\ 0 & 0 & m \end{array}\right]$$

Left: factor membership matrix L (assuming all β's are equal to 1). Right: lag matrix, capturing the (nonzero) lag of each stock to its corresponding factor.

Single membership model with $k = 3$ factors/signals

$$L = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{array}\right] \qquad G = \left[\begin{array}{ccc} 3 & 0 & 0 \\ 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ m & 0 & 0 \\ 2 & 0 & 0 \\ \hline 0 & 5 & 0 \\ 0 & m & 0 \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \\ 0 & 3 & 0 \\ \hline 0 & 0 & 3 \\ 0 & 0 & 2 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 4 \\ 0 & 0 & m \end{array}\right]$$

Left: factor membership matrix L (assuming all β's are equal to 1). Right: lag matrix, capturing the (nonzero) lag of each stock to its corresponding factor.

Ongoing work: employ clustering to propose an algorithm that recovers the matrix G, and leverage this for prediction tasks.

local2global cluster-driven methodology

- input: set of n time series each of length T, stacked as $X_{n \times T}$.

local2global cluster-driven methodology

- input: set of n time series each of length T, stacked as $X_{n \times T}$.
- extract subsequence time series (STS) of length q from each time series X_i by a sliding window

local2global cluster-driven methodology

- input: set of n time series each of length T, stacked as $X_{n\times T}$.
- extract subsequence time series (STS) of length q from each time series X_i by a sliding window
- enlarged universe matrix $U_{N\times q}$ is constructed by putting all STS together in order.

local2global cluster-driven methodology

- input: set of n time series each of length T, stacked as $X_{n\times T}$.
- extract subsequence time series (STS) of length q from each time series X_i by a sliding window
- enlarged universe matrix $U_{N\times q}$ is constructed by putting all STS together in order.

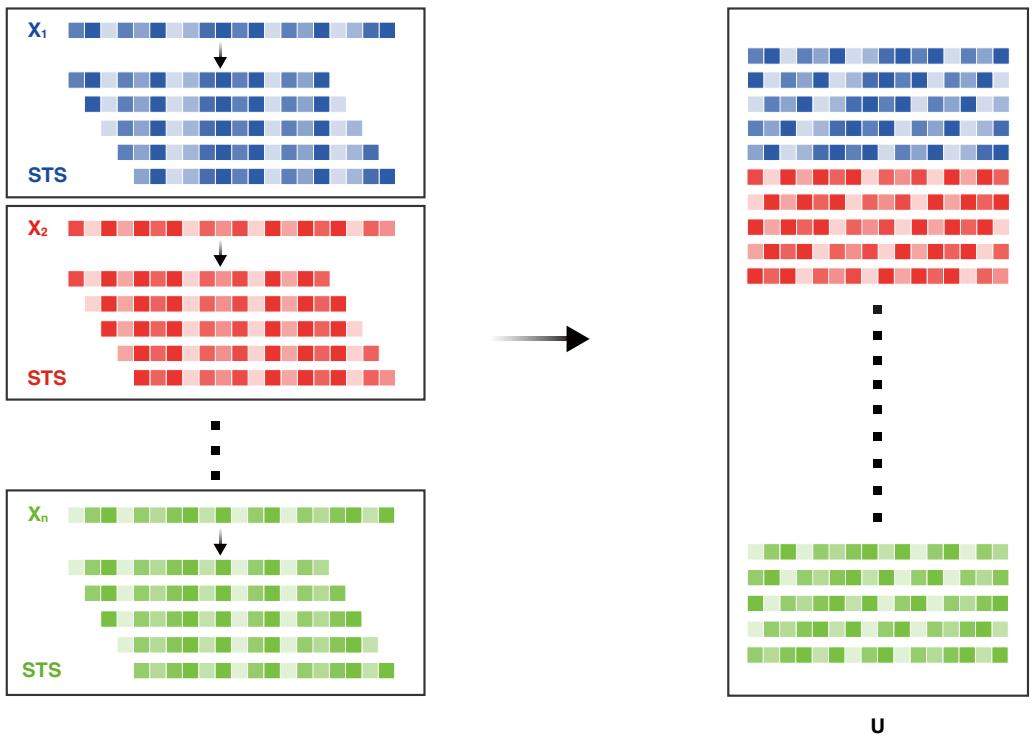


Figure: Left: STS are extracted from each time series by a sliding window. Right: An enlarged universe matrix U

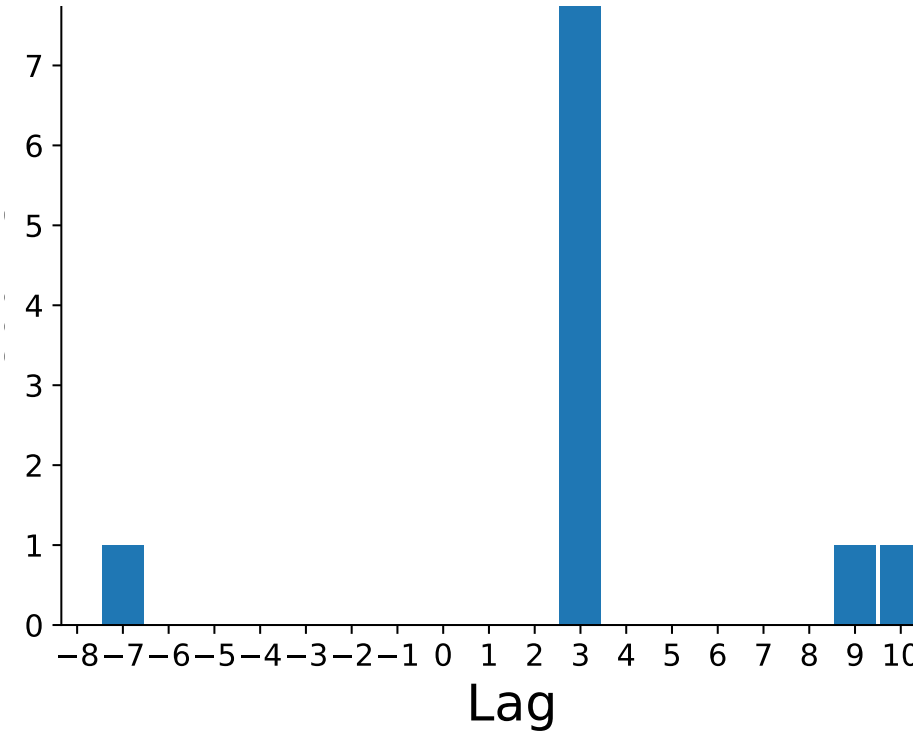
Zhang, C., Shestopaloff, Zohren, *Robust Detection of Lead-Lag Relationships in Lagged Multi-Factor Models* (2023) https://arxiv.org/abs/2305.06704

Methodology

Example: Y_i^t denotes the substring time series (STS) corresponding to stock i starting at day t.

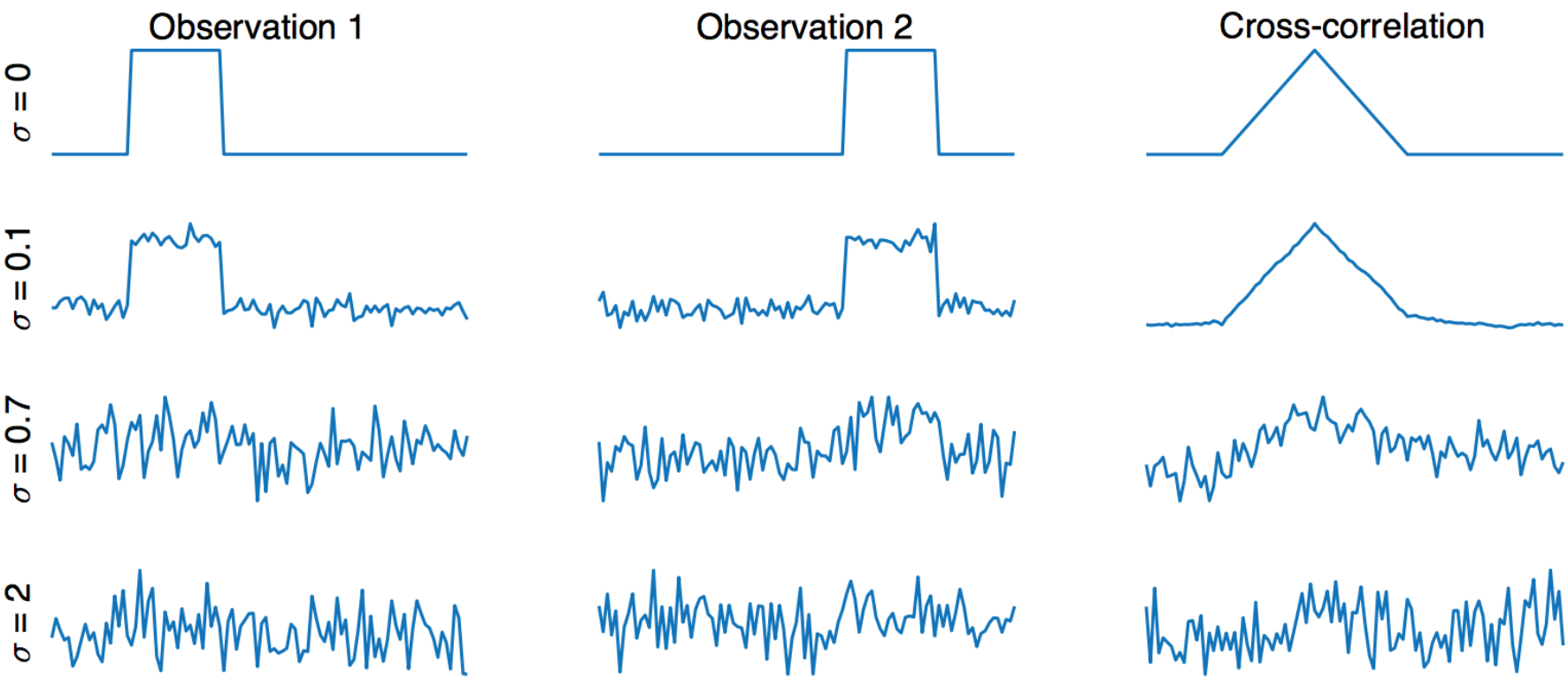
Cluster	Subsequence	Lag
ϕ_1	(Y_1^9, Y_2^2)	-7
ϕ_2	(Y_1^7, Y_2^{10})	3
ϕ_3	(Y_1^1, Y_2^4)	3
ϕ_4	(Y_1^2, Y_2^5)	3
ϕ_5	(Y_1^0, Y_2^3)	3
ϕ_6	(Y_1^6, Y_2^9)	3
ϕ_7	(Y_1^3, Y_2^6)	3
ϕ_8	(Y_1^4, Y_2^7)	3
ϕ_9	(Y_1^5, Y_2^8)	3
ϕ_{10}	$(Y_1^10, Y_2^0), (Y_1^10, Y_2^1)$	-10, -9
ϕ_{11}	Y_1^8	NaN

Example of calculating the relative lags of STS (substring time series) in each cluster from two time series.



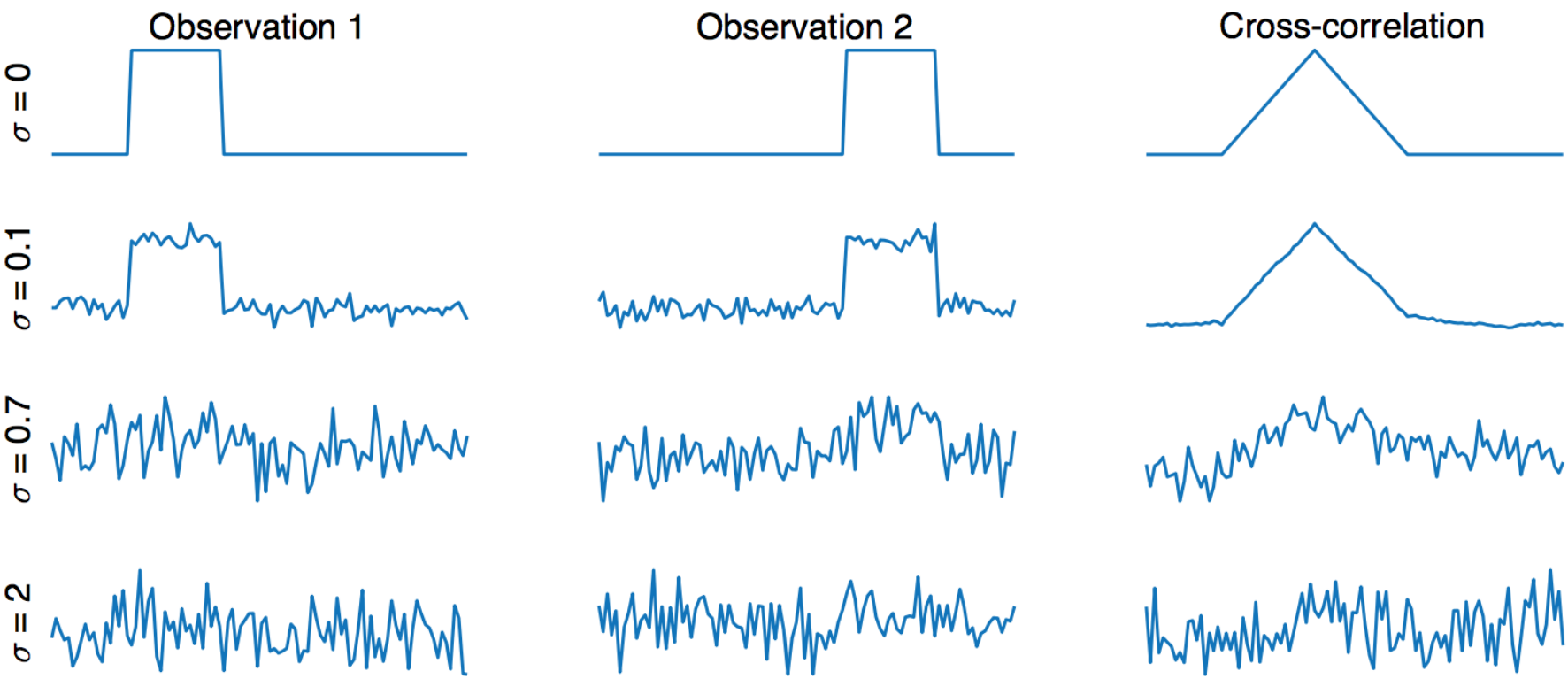
Histogram of the relative lags of STS from two time series.

Multi-reference alignment



From: T. Bendory, N. Boumal, C. Ma, Z. Zhao, A. Singer, ”Bispectrum Inversion with Application to Multireference Alignment" (2016)

Multi-reference alignment



From: T. Bendory, N. Boumal, C. Ma, Z. Zhao, A. Singer, ”Bispectrum Inversion with Application to Multireference Alignment" (2016)

Ongoing work: detect lags based on invariant feature representations and SDP and non-convex optimization.

Summary and outlook

Spectral methods:

- computational scalability
- robust to high level of noise in the data (low-SNR regime)
- theoretical guarantees under suitably defined stochastic block models

Summary and outlook

Spectral methods:

- computational scalability
- robust to high level of noise in the data (low-SNR regime)
- theoretical guarantees under suitably defined stochastic block models

Clustering:

- unsupervised ML provides insights into the structure of financial markets
- could be leveraged for downstream task of interest (lead-lag detection)

Summary and outlook

Spectral methods:

- computational scalability
- robust to high level of noise in the data (low-SNR regime)
- theoretical guarantees under suitably defined stochastic block models

Clustering:

- unsupervised ML provides insights into the structure of financial markets
- could be leveraged for downstream task of interest (lead-lag detection)

Main takeaways:

- best not to use off-the-shelf generic clustering algorithms
- it pays off to customize to the task at hand (task-driven objective function)

Summary and outlook

Spectral methods:

- computational scalability
- robust to high level of noise in the data (low-SNR regime)
- theoretical guarantees under suitably defined stochastic block models

Clustering:

- unsupervised ML provides insights into the structure of financial markets
- could be leveraged for downstream task of interest (lead-lag detection)

Main takeaways:

- best not to use off-the-shelf generic clustering algorithms
- it pays off to customize to the task at hand (task-driven objective function)

Future & ongoing work:

- explore additional data sets - higher frequency setting (lead-lag)

Summary and outlook

Spectral methods:

- computational scalability
- robust to high level of noise in the data (low-SNR regime)
- theoretical guarantees under suitably defined stochastic block models

Clustering:

- unsupervised ML provides insights into the structure of financial markets
- could be leveraged for downstream task of interest (lead-lag detection)

Main takeaways:

- best not to use off-the-shelf generic clustering algorithms
- it pays off to customize to the task at hand (task-driven objective function)

Future & ongoing work:

- explore additional data sets - higher frequency setting (lead-lag)
- leverage clustering in statistical arbitrage settings, correlation/covariance estimation, risk model construction

Summary and outlook

Spectral methods:

- computational scalability
- robust to high level of noise in the data (low-SNR regime)
- theoretical guarantees under suitably defined stochastic block models

Clustering:

- unsupervised ML provides insights into the structure of financial markets
- could be leveraged for downstream task of interest (lead-lag detection)

Main takeaways:

- best not to use off-the-shelf generic clustering algorithms
- it pays off to customize to the task at hand (task-driven objective function)

Future & ongoing work:

- explore additional data sets - higher frequency setting (lead-lag)
- leverage clustering in statistical arbitrage settings, correlation/covariance estimation, risk model construction
- extension to the time dependent setting - what are good probabilistic models to track temporal changes in a network?

Summary and outlook

Spectral methods:

- computational scalability
- robust to high level of noise in the data (low-SNR regime)
- theoretical guarantees under suitably defined stochastic block models

Clustering:

- unsupervised ML provides insights into the structure of financial markets
- could be leveraged for downstream task of interest (lead-lag detection)

Main takeaways:

- best not to use off-the-shelf generic clustering algorithms
- it pays off to customize to the task at hand (task-driven objective function)

Future & ongoing work:

- explore additional data sets - higher frequency setting (lead-lag)
- leverage clustering in statistical arbitrage settings, correlation/covariance estimation, risk model construction
- extension to the time dependent setting - what are good probabilistic models to track temporal changes in a network?
- change-point detection in (signed/directed) network time series data

Summary and outlook

Spectral methods:

- computational scalability
- robust to high level of noise in the data (low-SNR regime)
- theoretical guarantees under suitably defined stochastic block models

Clustering:

- unsupervised ML provides insights into the structure of financial markets
- could be leveraged for downstream task of interest (lead-lag detection)

Main takeaways:

- best not to use off-the-shelf generic clustering algorithms
- it pays off to customize to the task at hand (task-driven objective function)

Future & ongoing work:

- explore additional data sets - higher frequency setting (lead-lag)
- leverage clustering in statistical arbitrage settings, correlation/covariance estimation, risk model construction
- extension to the time dependent setting - what are good probabilistic models to track temporal changes in a network?
- change-point detection in (signed/directed) network time series data
- extension to hypergraphs and higher-order structures

Summary and outlook

Spectral methods:

- computational scalability
- robust to high level of noise in the data (low-SNR regime)
- theoretical guarantees under suitably defined stochastic block models

Clustering:

- unsupervised ML provides insights into the structure of financial markets
- could be leveraged for downstream task of interest (lead-lag detection)

Main takeaways:

- best not to use off-the-shelf generic clustering algorithms
- it pays off to customize to the task at hand (task-driven objective function)

Future & ongoing work:

- explore additional data sets - higher frequency setting (lead-lag)
- leverage clustering in statistical arbitrage settings, correlation/covariance estimation, risk model construction
- extension to the time dependent setting - what are good probabilistic models to track temporal changes in a network?
- change-point detection in (signed/directed) network time series data
- extension to hypergraphs and higher-order structures
- detection of short-lived localized correlations clusters in large baskets

Summary and outlook

Spectral methods:

- computational scalability
- robust to high level of noise in the data (low-SNR regime)
- theoretical guarantees under suitably defined stochastic block models

Clustering:

- unsupervised ML provides insights into the structure of financial markets
- could be leveraged for downstream task of interest (lead-lag detection)

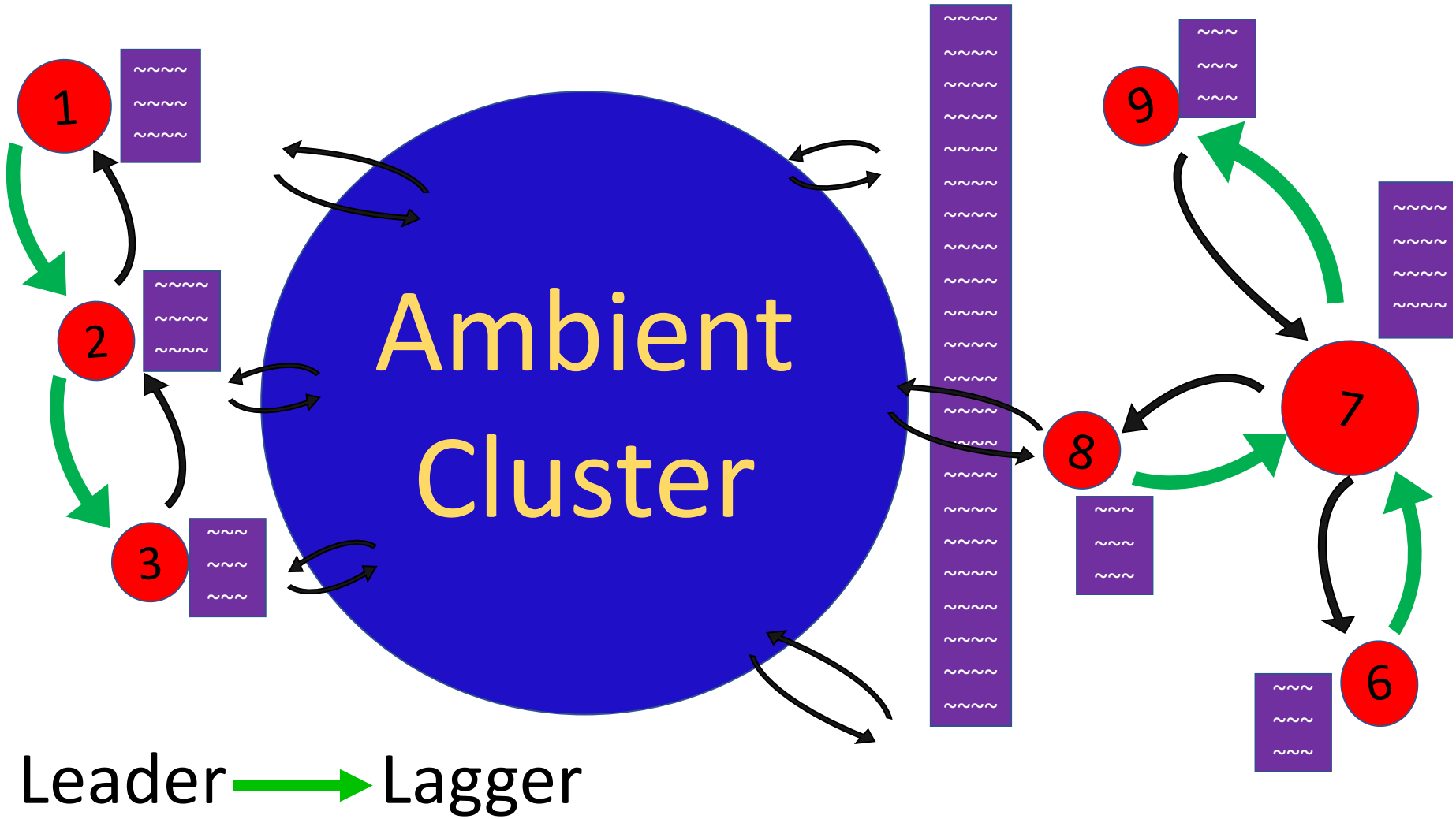
Main takeaways:

- best not to use off-the-shelf generic clustering algorithms
- it pays off to customize to the task at hand (task-driven objective function)

Future & ongoing work:

- explore additional data sets - higher frequency setting (lead-lag)
- leverage clustering in statistical arbitrage settings, correlation/covariance estimation, risk model construction
- extension to the time dependent setting - what are good probabilistic models to track temporal changes in a network?
- change-point detection in (signed/directed) network time series data
- extension to hypergraphs and higher-order structures
- detection of short-lived localized correlations clusters in large baskets
- graph neural network (GNN) for signed/directed/time series networks

Planted structure in a much larger ambient graph + Handling node level covariates



Motivates the use Graph Neural Networks for digraph clustering (ongoing work).

Regularization in the sparse regime

- adding a regularization step was shown to significantly improve the performance of spectral algorithms in the very sparse regime $p = \Theta(\frac{1}{n})$

Regularization in the sparse regime

- adding a regularization step was shown to significantly improve the performance of spectral algorithms in the very sparse regime $p = \Theta(\frac{1}{n})$
- Joseph and Yu (2016) provided a theoretical justification for the regularization $A_\tau = A + \tau \mathbf{1}\mathbf{1}^T$ (**1** denotes the all ones column vector)

Regularization in the sparse regime

- adding a regularization step was shown to significantly improve the performance of spectral algorithms in the very sparse regime $p = \Theta(\frac{1}{n})$
- Joseph and Yu (2016) provided a theoretical justification for the regularization $A_\tau = A + \tau \mathbf{1}\mathbf{1}^T$ (**1** denotes the all ones column vector)
- a *signed regularization* step amounts to adding a weight to each edge (including self-loops) of the positive and negative subgraphs

Regularization in the sparse regime

- adding a regularization step was shown to significantly improve the performance of spectral algorithms in the very sparse regime $p = \Theta(\frac{1}{n})$
- Joseph and Yu (2016) provided a theoretical justification for the regularization $A_\tau = A + \tau \mathbf{1}\mathbf{1}^T$ ($\mathbf{1}$ denotes the all ones column vector)
- a *signed regularization* step amounts to adding a weight to each edge (including self-loops) of the positive and negative subgraphs
- for some regularization parameters $\gamma^+, \gamma^- \geq 0$, define

$$A^+_{\gamma^+} := A^+ + \frac{\gamma^+}{n}\mathbf{1}\mathbf{1}^\top; \qquad A^-_{\gamma^-} := A^- + \frac{\gamma^-}{n}\mathbf{1}\mathbf{1}^\top$$

the regularized adjacency matrices for the unsigned graphs G^+, G^-

Regularization in the sparse regime

- adding a regularization step was shown to significantly improve the performance of spectral algorithms in the very sparse regime $p = \Theta(\frac{1}{n})$
- Joseph and Yu (2016) provided a theoretical justification for the regularization $A_\tau = A + \tau \mathbf{1}\mathbf{1}^T$ ($\mathbf{1}$ denotes the all ones column vector)
- a *signed regularization* step amounts to adding a weight to each edge (including self-loops) of the positive and negative subgraphs
- for some regularization parameters $\gamma^+, \gamma^- \geq 0$, define

$$A^+_{\gamma^+} := A^+ + \frac{\gamma^+}{n}\mathbf{1}\mathbf{1}^\top; \qquad A^-_{\gamma^-} := A^- + \frac{\gamma^-}{n}\mathbf{1}\mathbf{1}^\top$$

  the regularized adjacency matrices for the unsigned graphs G^+, G^-
- $L^\pm_{sym,\gamma^\pm}$ the normalized Laplacians corresponding to $A^\pm_{\gamma^\pm}$

Regularization in the sparse regime

- adding a regularization step was shown to significantly improve the performance of spectral algorithms in the very sparse regime $p = \Theta(\frac{1}{n})$
- Joseph and Yu (2016) provided a theoretical justification for the regularization $A_\tau = A + \tau \mathbf{1}\mathbf{1}^T$ ($\mathbf{1}$ denotes the all ones column vector)
- a *signed regularization* step amounts to adding a weight to each edge (including self-loops) of the positive and negative subgraphs
- for some regularization parameters $\gamma^+, \gamma^- \geq 0$, define

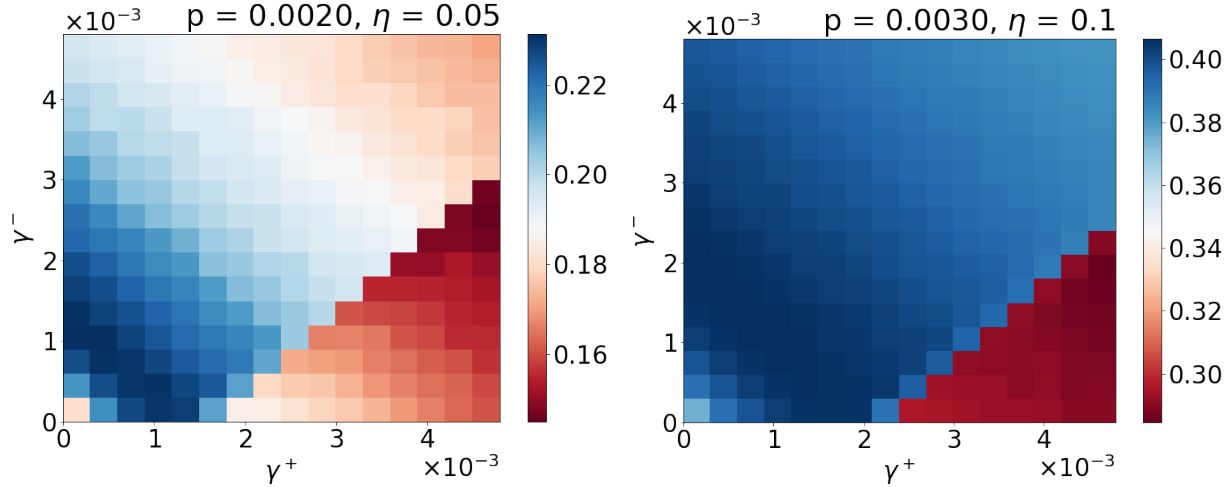
$$A^+_{\gamma^+} := A^+ + \frac{\gamma^+}{n}\mathbf{1}\mathbf{1}^\top; \qquad A^-_{\gamma^-} := A^- + \frac{\gamma^-}{n}\mathbf{1}\mathbf{1}^\top$$

  the regularized adjacency matrices for the unsigned graphs G^+, G^-
- $L^\pm_{sym,\gamma^\pm}$ the normalized Laplacians corresponding to $A^\pm_{\gamma^\pm}$
- finally, denote L_γ the regularized symmetric Signed Laplacian

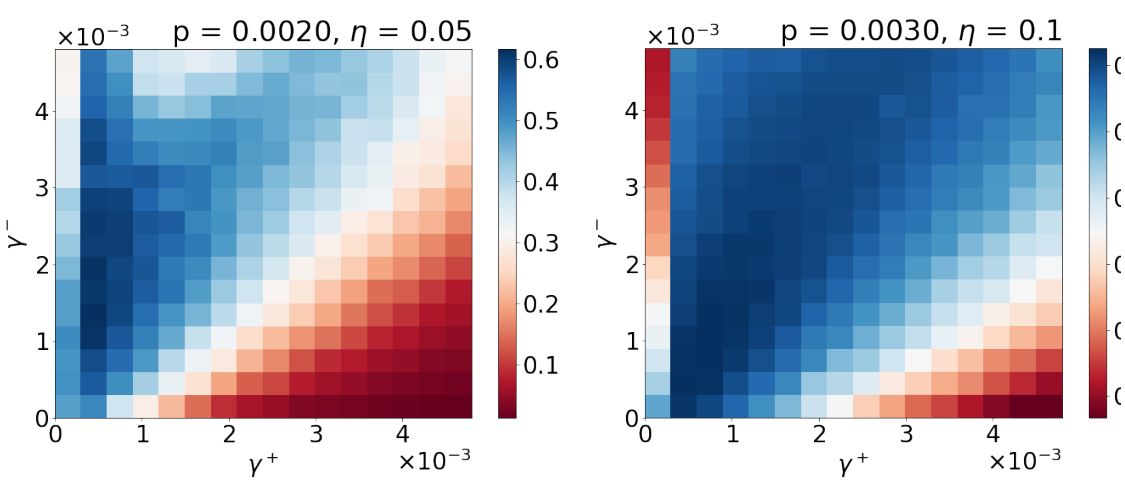
$$L_\gamma = I - (\overline{D}_\gamma)^{-1/2} A_\gamma (\overline{D}_\gamma)^{-1/2}, \tag{31}$$

with $A_\gamma = A^+_{\gamma^+} - A^-_{\gamma^-}$ and $\overline{D}_\gamma = \overline{D} + (\gamma^+ + \gamma^-)I$.

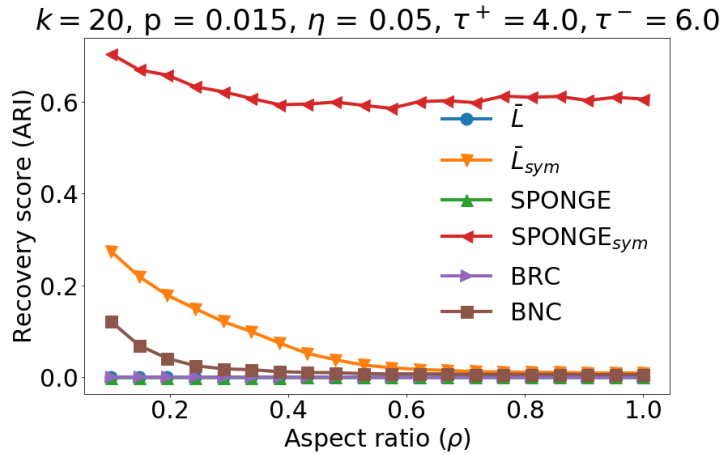
Regularization in the sparse SSBM



(a) Regularized Signed Laplacian L_γ



(b) SPONGE$_{sym}$



(c) Comparison

Figure: Heatmaps of the ARI obtained with the two sparse algorithms, L_γ and SPONGE$_{sym}$, with varying regularization parameters (γ^+, γ^-), for a SSBM in two **sparse** regimes, with $n = 5000$ and $k = 5$ clusters. Plot (c) $k = 20$.