Network analysis in financial applications and interplay with time series data

OxML x Finance Summer School 2022

Mihai Cucuringu
mihai.cucuringu@stats.ox.ac.uk

University of Oxford

Brief intro to network analysis

Signed graph clustering

- Application to news sentiment propagation

Directed graph clustering

Further topics within financial networks

Leaders and laggers in time series data

Price Impact of Order Flow Imbalance: Multi-level, Cross-asset and Forecasting

Introduction

- Networks are just graphs (we shall use *network* and *graph* interchangeably)

Introduction

- Networks are just graphs (we shall use *network* and *graph* interchangeably)
- Networks provide a useful representation of interdependencies in data

Introduction

- Networks are just graphs (we shall use *network* and *graph* interchangeably)
- Networks provide a useful representation of interdependencies in data
- Networks are also used to represent statistical models - so-called graphical models - but this lecture does not address graphical models

Network of financial assets

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

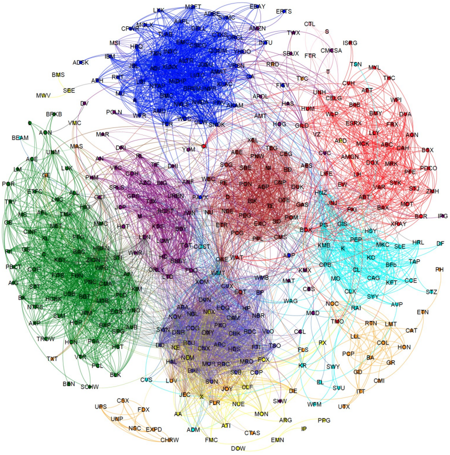
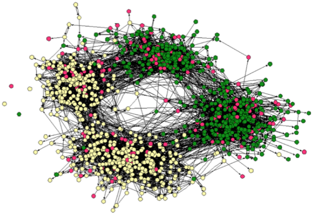
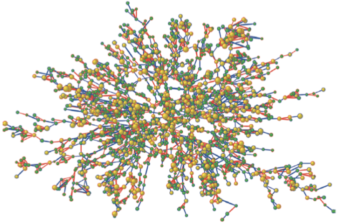


Figure: Asset correlation matrix after thresholding. The color of each node represents the industry sector to which that stock belongs. The force-based layout clearly indicates the existence of strong connections between stocks of the same industry sector.

Analysis of graph data sets in the past

- the study of networks has a long tradition originating in social science (*Social Network Analysis*)
- the networks under consideration are typically fairly small
- visual inspection can reveal a lot of information





Analysis of graph data sets now

In contrast, starting at around 1997, **statistical physicists** have turned their attention to large-scale properties of networks.

Analysis of graph data sets now

In contrast, starting at around 1997, **statistical physicists** have turned their attention to large-scale properties of networks.

- unless the network is very small it appears like a hairball, and is difficult to analyse by just looking at it

Analysis of graph data sets now

In contrast, starting at around 1997, **statistical physicists** have turned their attention to large-scale properties of networks.

- unless the network is very small it appears like a hairball, and is difficult to analyse by just looking at it
- more and larger networks appear (byproducts of technological advancement)
  - e.g., internet, web
  - result of our ability to collect more, better-quality, and more complex data

Analysis of graph data sets now

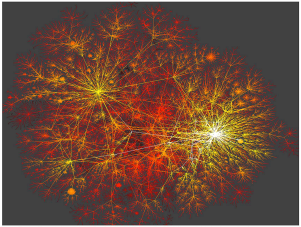
In contrast, starting at around 1997, **statistical physicists** have turned their attention to large-scale properties of networks.

- unless the network is very small it appears like a hairball, and is difficult to analyse by just looking at it
- more and larger networks appear (byproducts of technological advancement)
  - e.g., internet, web
  - result of our ability to collect more, better-quality, and more complex data
- networks of thousands, millions, or billions of nodes
  - in financial applications, banks often have millions of accounts, the network captures the flow of financial transactions

Analysis of graph data sets now

In contrast, starting at around 1997, **statistical physicists** have turned their attention to large-scale properties of networks.

- unless the network is very small it appears like a hairball, and is difficult to analyse by just looking at it
- more and larger networks appear (byproducts of technological advancement)
  - e.g., internet, web
  - result of our ability to collect more, better-quality, and more complex data
- networks of thousands, millions, or billions of nodes
  - in financial applications, banks often have millions of accounts, the network captures the flow of financial transactions
- need to develop more sophisticated & scalable tools



Frieze, Gionis, Tsourakakis   Algorithmic Techniques for Modeling and Mining Large Graphs

Figure: The Internet graph.

Network Science

Emerging area with an exponential growth of its literature over the past decades.

Network Science

Emerging area with an exponential growth of its literature over the past decades.

The world is full with networks - what do we do with them?

- understand their topology and measure their properties

Network Science

Emerging area with an exponential growth of its literature over the past decades.

The world is full with networks - what do we do with them?

- understand their topology and measure their properties
- study their evolution and dynamics

Network Science

Emerging area with an exponential growth of its literature over the past decades.

The world is full with networks - what do we do with them?

- understand their topology and measure their properties
- study their evolution and dynamics
- create realistic models

Network Science

Emerging area with an exponential growth of its literature over the past decades.

The world is full with networks - what do we do with them?

- understand their topology and measure their properties
- study their evolution and dynamics
- create realistic models
- build efficient algorithms that can leverage the network structure

Network Science

Emerging area with an exponential growth of its literature over the past decades.

The world is full with networks - what do we do with them?

- understand their topology and measure their properties
- study their evolution and dynamics
- create realistic models
- build efficient algorithms that can leverage the network structure
- perform various tasks on the network (e.g., ranking, prediction)

Network Science

Emerging area with an exponential growth of its literature over the past decades.

The world is full with networks - what do we do with them?

- understand their topology and measure their properties
- study their evolution and dynamics
- create realistic models
- build efficient algorithms that can leverage the network structure
- perform various tasks on the network (e.g., ranking, prediction)

Network Science

Emerging area with an exponential growth of its literature over the past decades.

The world is full with networks - what do we do with them?

- understand their topology and measure their properties
- study their evolution and dynamics
- create realistic models
- build efficient algorithms that can leverage the network structure
- perform various tasks on the network (e.g., ranking, prediction)

Question: how can we leverage structural findings in a network for prediction? (Also allows for a fair comparison of different methods.)

Further research questions

- How do these networks work? Where could we best manipulate a network in order to prevent, say, tumor growth?

Further research questions

- How do these networks work? Where could we best manipulate a network in order to prevent, say, tumor growth?

- How did these networks evolve?

Further research questions

- How do these networks work? Where could we best manipulate a network in order to prevent, say, tumor growth?
- How did these networks evolve?
- How similar are these networks?

Further research questions

- How do these networks work? Where could we best manipulate a network in order to prevent, say, tumor growth?
- How did these networks evolve?
- How similar are these networks?
- How are these networks interlinked?

Further research questions

- How do these networks work? Where could we best manipulate a network in order to prevent, say, tumor growth?
- How did these networks evolve?
- How similar are these networks?
- How are these networks interlinked?
- What are the building principles of these networks? How is resilience achieved, and how is flexibility achieved?

From a statistical viewpoint, questions include

- How to best describe networks?

From a statistical viewpoint, questions include

- How to best describe networks?
- How to infer characteristics of vertices in the network?

From a statistical viewpoint, questions include

- How to best describe networks?
- How to infer characteristics of vertices in the network?
- How to infer missing links, and how to check whether existing links are not false positives?

From a statistical viewpoint, questions include

- How to best describe networks?
- How to infer characteristics of vertices in the network?
- How to infer missing links, and how to check whether existing links are not false positives?
- How to compare networks? (use case in change-point detection)

From a statistical viewpoint, questions include

- How to best describe networks?
- How to infer characteristics of vertices in the network?
- How to infer missing links, and how to check whether existing links are not false positives?
- How to compare networks? (use case in change-point detection)

From a statistical viewpoint, questions include

- How to best describe networks?
- How to infer characteristics of vertices in the network?
- How to infer missing links, and how to check whether existing links are not false positives?
- How to compare networks? (use case in change-point detection)
- How to find relevant sub-structures of a network?

From a statistical viewpoint, questions include

- How to best describe networks?
- How to infer characteristics of vertices in the network?
- How to infer missing links, and how to check whether existing links are not false positives?
- How to compare networks? (use case in change-point detection)
- How to find relevant sub-structures of a network?

• Statistical inference relies on the assumption that there is some randomness in the data.

Clustering graphs

- Consider an undirected graph $G = (V, E)$ with n vertices

Clustering graphs

- Consider an undirected graph $G = (V, E)$ with n vertices
- Each $\{i, j\} \in E$ has an associated **positive** weight $w_{ij} \geq 0$ (similarity between vertices)

Clustering graphs

- Consider an undirected graph $G = (V, E)$ with n vertices
- Each $\{i, j\} \in E$ has an associated **positive** weight $w_{ij} \geq 0$ (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 $w_{ij} \geq 0$ (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 $w_{ij} \geq 0$ (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.

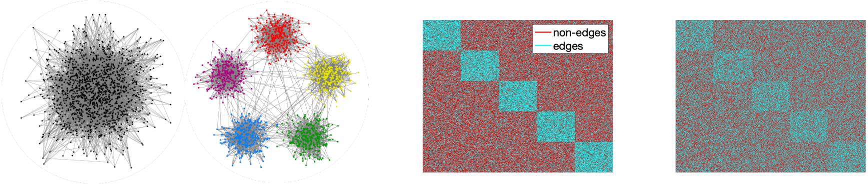


Figure: [Abbe '17] Recover the right graph from the left (scrambled) graph.

Clustering graphs

- Consider an undirected graph $G = (V, E)$ with n vertices
- Each $\{i, j\} \in E$ has an associated **positive** weight $w_{ij} \geq 0$ (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.

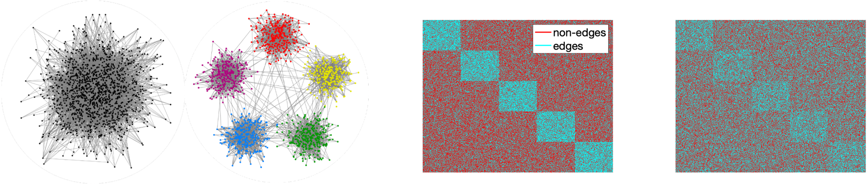


Figure: [Abbe '17] Recover the right graph from the left (scrambled) graph.

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

- the sign or directionality of the edges
- the signals sitting at the nodes.

Graph Cuts

- consider a partition of the graph G into two subgraphs A and B

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

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

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

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

- 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

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

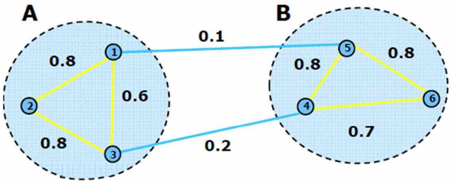
- the notion of a **cut** is a fundamental concept in graph clustering
- we aim to find a partition/split of G into A and B in order to minimize the resulting cut.

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

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

- the notion of a **cut** is a fundamental concept in graph clustering
- we aim to find a partition/split of G into A and B in order to minimize the resulting cut.



[Source: David Sontag]

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

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

- 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

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

- 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 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 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 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.

Spectral clustering of 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 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 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 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

$$x^T L\, x = \sum_{(i,j)\in E} (x_i - x_j)^2$$

Signed Graphs and Signed Graph Cuts

- Many applications involve graphs where edge weights can take negative values (dissimilarity) as well

Signed Graphs and Signed Graph Cuts

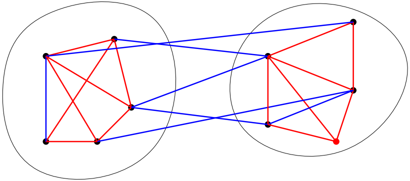
- Many applications involve graphs where edge weights can take negative values (dissimilarity) as well
  - **Social networks:** users are friends ('+$'$ edge) or enemies ('-' edge)

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

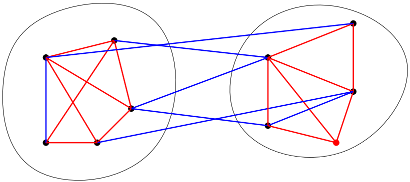
- Many applications involve graphs where edge weights can take negative values (dissimilarity) as well
  - **Social networks:** users are friends ('+' edge) or enemies ('-' edge)
  - **Image segmentation:** edges connect adjacent pixels and encode (dis-)similarity information
  - **Time series clustering**: 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 sum of weights of: intra cluster + edges plus inter cluster – edges, or
- Minimize sum of weights of: inter cluster + edges plus intra cluster – edges

Further motivation: Statistical Arbitrage

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 information technology infrastructure

Further motivation: Statistical Arbitrage

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 information 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

Further motivation: Statistical Arbitrage

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 information 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.

Network modeling methods for FMRI. Neuroimage (2011)

- detailed survey & comparison of methods for turning time series into networks
- interplay between fMRI time series and the network generation process

Neuroimage. 2011 Jan 15;54(2):875-91. doi: 10.1016/j.neuroimage.2010.08.063. Epub 2010 Sep 15.

Network modelling methods for FMRI.

Smith SM[1], Miller KL, Salimi-Khorshidi G, Webster M, Beckmann CF, Nichols TE, Ramsey JD, Woolrich MW.

Author information

1 FMRIB (Oxford University Centre for Functional MRI of the Brain), Department of Clinical Neurology, Uni
steve@fmrib.ox.ac.uk

Abstract

There is great interest in estimating brain "networks" from FMRI data. This is often attempted by identi
(e.g., spatial ROIs or ICA maps) and then conducting a connectivity analysis between the nodes, base
associated with the nodes. Analysis methods range from very simple measures that consider just two
between two nodes' timeseries) to sophisticated approaches that consider all nodes simultaneously ar
model (e.g., Bayes net models). Many different methods are being used in the literature, but almost nc
compared for use on FMRI timeseries data. In this work we generate rich, realistic simulated FMRI dat
networks, experimental protocols and problematic confounds in the data, in order to compare different
approaches. Our results show that in general correlation-based approaches can be quite successful, r
statistics are less sensitive, and lag-based approaches perform very poorly. More specifically: there ar
high sensitivity to network connection detection on good quality FMRI data, in particular, partial correla
covariance estimation and several Bayes net methods; however, accurate estimation of connection dir
achieve, though Patel's τ can be reasonably successful. With respect to the various confounds added
was that the use of functionally inaccurate ROIs (when defining the network nodes and extracting their
extremely damaging to network estimation; hence, results derived from inappropriate ROI definition (s
should be regarded with great caution.

From time series to networks

- *"Detecting a currency's dominance or dependence using foreign exchange network trees"*, McDonald et al., Phys. Rev. E 72, 046106, 2005
  - a network analysis of correlations (from the FX market) using minimum spanning trees (MSTs)
  - show MSTs provide a meaningful representation of the global FX dynamics & allow to determine momentarily dominant and dependent currencies
  - to construct the MST, they convert the Pearson correlation matrix $\rho_{n\times n}$ into a "distance" matrix D, via the nonlinear mapping/ultrametric distance

$$d_{ij}(\rho_{ij}) = \sqrt{2(1-\rho_{ij})} \tag{4}$$

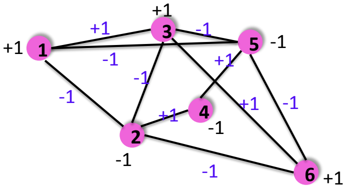
- *"Phase separation and scaling in correlation structures of financial markets"*, Chakraborti et al., Journal of Physics: Complexity 2.1 (2020): 015002.
  - construct weighted graphs by powering the absolute values of the correlation coefficients (default power value is 2)
- *"Network geometry & market instability"*, Samal et al., Royal Soc. Open Sc. 2021
  - transform the correlation into the ultrametric distance $D(t)_{ij} = \sqrt{2(1-\rho_{ij}}$, then find a MST on the resulting weighted complete graph and finally re-include edges such that $\rho_{ij} > 0.75$.
- other distances used:

$$d^{(1)}_{ij}(\rho_{ij}) = 1 - |\rho_{ij}| \tag{5}$$

$$d^{(2)}_{ij}(\rho_{ij}) = 1 - |\rho_{ij}|^2 \tag{6}$$

- but what if we want to work directly with the correlations?

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



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

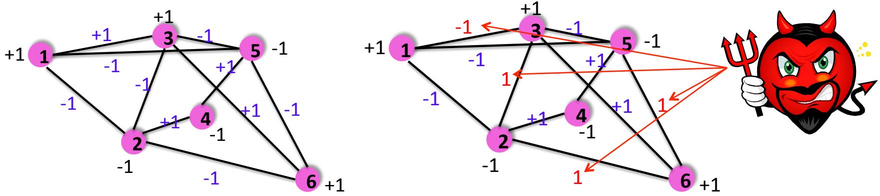


Figure: Synchronization over $\mathbb{Z}_2$ (left: clean, right: noisy)

- unknown group elements $z_1, z_2, \dots, z_N \in \mathbb{Z}_2$ (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 $k = 2$ (Synchronization over $\mathbb{Z}_2$)

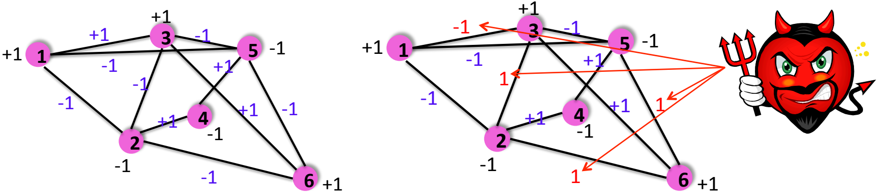


Figure: Synchronization over $\mathbb{Z}_2$ (left: clean, right: noisy)

- 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

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

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

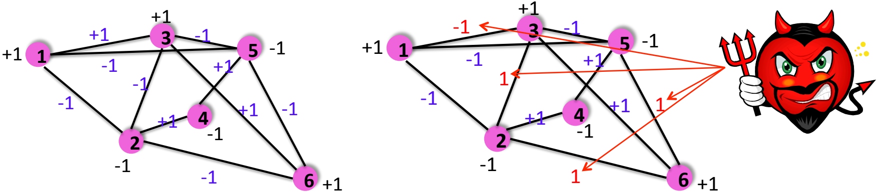


Figure: Synchronization over $\mathbb{Z}_2$ (left: clean, right: noisy)

- 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

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

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

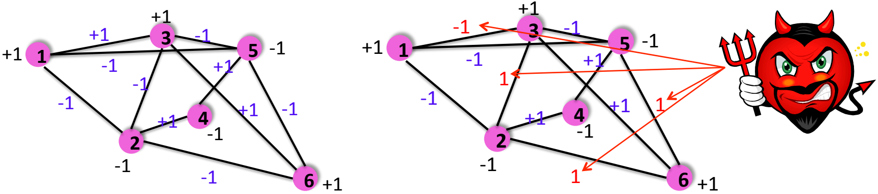


Figure: Synchronization over $\mathbb{Z}_2$ (left: clean, right: noisy)

- 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

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

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

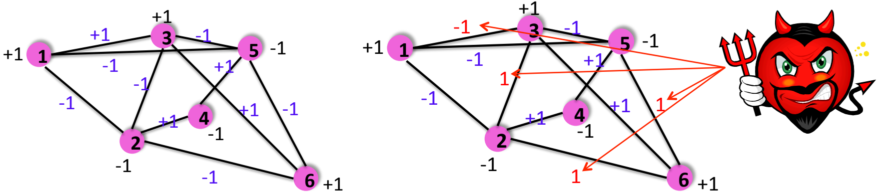


Figure: Synchronization over $\mathbb{Z}_2$ (left: clean, right: noisy)

- 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

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

- original solution: $z_1, \ldots, z_n \in \pm 1^n \quad (\mathbb{Z}_2 = \{-1, +1\})$

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

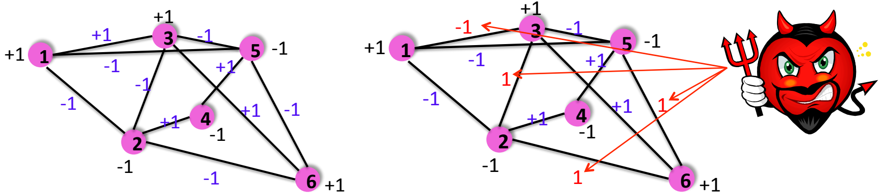


Figure: Synchronization over $\mathbb{Z}_2$ (left: clean, right: noisy)

- 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

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

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

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

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

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

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

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

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

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

whose maximum is attained when $x = z$ (noise-free data).

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

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

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

whose maximum is attained when $x = z$ (noise-free data).

- NP-hard problem, but relax (drop $\pm$ constraints) to

$$\max_{\sum_{i=1}^{N} |x_i|^2 = N} \sum_{i,j=1}^{N} x_i A_{ij} x_j$$

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

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

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

whose maximum is attained when $x = z$ (noise-free data).

- NP-hard problem, but relax (drop $\pm$ constraints) to

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

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

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

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

whose maximum is attained when $x = z$ (noise-free data).

- NP-hard problem, but relax (drop $\pm$ constraints) to

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

whose maximum is achieved when

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

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

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

whose maximum is attained when $x = z$ (noise-free data).

- NP-hard problem, but relax (drop $\pm$ constraints) to

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

whose maximum is achieved when $x = v_1$, the normalized top eigenvector of A that solves

$$A v_1 = \lambda_1 v_1$$

Alternative formulation - Synchronization over $\mathbb{Z}_2$

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

$$\min_{x \in \mathbb{Z}_2^N} \sum_{(i,j) \in E} \left(x_i - A_{ij} x_j \right)^2$$

Alternative formulation - Synchronization over $\mathbb{Z}_2$

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

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

Alternative formulation - Synchronization over $\mathbb{Z}_2$

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

$$\begin{aligned} \min_{x \in \mathbb{Z}_2^N} \sum_{(i,j) \in E} (x_i - A_{ij} x_j)^2 &= \min_{x \in \mathbb{Z}_2^N} \sum_{(i,j) \in E} x_i^2 + A_{ij}^2 x_j^2 - 2 A_{ij} x_i x_j \\ &= \min_{x \in \mathbb{Z}_2^N} \sum_{(i,j) \in E} x_i^2 + x_j^2 - 2 A_{ij} x_i x_j \end{aligned}$$

Alternative formulation - Synchronization over $\mathbb{Z}_2$

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

$$
\begin{aligned}
\min_{x \in \mathbb{Z}_2^N} \sum_{(i,j) \in E} (x_i - A_{ij} x_j)^2 &= \min_{x \in \mathbb{Z}_2^N} \sum_{(i,j) \in E} x_i^2 + A_{ij}^2 x_j^2 - 2A_{ij} x_i x_j \\
&= \min_{x \in \mathbb{Z}_2^N} \sum_{(i,j) \in E} x_i^2 + x_j^2 - 2A_{ij} x_i x_j \\
&= \min_{x \in \mathbb{Z}_2^N} \sum_{i=1}^{n} d_i x_i^2 - \sum_{(i,j) \in E} 2A_{ij} x_i x_j
\end{aligned}
$$

Alternative formulation - Synchronization over $\mathbb{Z}_2$

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

$$
\begin{aligned}
\min_{x \in \mathbb{Z}_2^N} \sum_{(i,j)\in E} (x_i - A_{ij} x_j)^2 &= \min_{x \in \mathbb{Z}_2^N} \sum_{(i,j)\in E} x_i^2 + A_{ij}^2 x_j^2 - 2A_{ij} x_i x_j \\
&= \min_{x \in \mathbb{Z}_2^N} \sum_{(i,j)\in E} x_i^2 + x_j^2 - 2A_{ij} x_i x_j \\
&= \min_{x \in \mathbb{Z}_2^N} \sum_{i=1}^{n} d_i x_i^2 - \sum_{(i,j)\in E} 2A_{ij} x_i x_j \\
&= \min_{x \in \mathbb{Z}_2^N} x^T \overline{D} x - x^T A x
\end{aligned}
$$

Alternative formulation - Synchronization over $\mathbb{Z}_2$

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

$$
\begin{aligned}
\min_{x \in \mathbb{Z}_2^N} \sum_{(i,j)\in E} (x_i - A_{ij}x_j)^2 &= \min_{x \in \mathbb{Z}_2^N} \sum_{(i,j)\in E} x_i^2 + A_{ij}^2 x_j^2 - 2A_{ij}x_i x_j \\
&= \min_{x \in \mathbb{Z}_2^N} \sum_{(i,j)\in E} x_i^2 + x_j^2 - 2A_{ij}x_i x_j \\
&= \min_{x \in \mathbb{Z}_2^N} \sum_{i=1}^{n} d_i x_i^2 - \sum_{(i,j)\in E} 2A_{ij}x_i x_j \\
&= \min_{x \in \mathbb{Z}_2^N} x^T \overline{D} x - x^T A x \\
&= \min_{x \in \mathbb{Z}_2^N} x^T (\overline{D} - A) x
\end{aligned}
$$

Alternative formulation - Synchronization over $\mathbb{Z}_2$

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

$$
\begin{aligned}
\min_{x \in \mathbb{Z}_2^N} \sum_{(i,j)\in E} (x_i - A_{ij}x_j)^2 &= \min_{x \in \mathbb{Z}_2^N} \sum_{(i,j)\in E} x_i^2 + A_{ij}^2 x_j^2 - 2A_{ij}x_i x_j \\
&= \min_{x \in \mathbb{Z}_2^N} \sum_{(i,j)\in E} x_i^2 + x_j^2 - 2A_{ij}x_i x_j \\
&= \min_{x \in \mathbb{Z}_2^N} \sum_{i=1}^{n} d_i x_i^2 - \sum_{(i,j)\in E} 2A_{ij}x_i x_j \\
&= \min_{x \in \mathbb{Z}_2^N} x^T \overline{D} x - x^T A x \\
&= \min_{x \in \mathbb{Z}_2^N} x^T (\overline{D} - A) x
\end{aligned}
$$

- Signed Graph Laplacian $\overline{L} = \overline{D} - A$

Brief intro to network analysis

Signed graph clustering

Application to news sentiment propagation

Directed graph clustering

Further topics within financial networks

Leaders and laggers in time series data

Price Impact of Order Flow Imbalance: Multi-level, Cross-asset and Forecasting

Thomson Reuters news sentiment data

	Asset 1				Asset 2				Asset ...				Asset 500			
Time stamp	Class	+	0	−	Class	+	0	−	Class	+	0	−	Class	+	0	−
dd/mm/yyyy 00:00:00	○	○	○	○	●	●	●	●	○	○	○	○	●	●	●	●
dd/mm/yyyy 00:00:00	●	●	●	●	●	●	●	●	○	○	○	○	○	○	○	○
dd/mm/yyyy 00:02:00	○	○	○	○	○	○	○	○	●	●	●	●	○	○	○	○
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
dd/mm/yyyy 09:30:00	○	○	○	○	●	●	●	●	●	●	●	●	○	○	○	○
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
dd/mm/yyyy 16:00:00	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●	●
dd/mm/yyyy 16:01:00	●	●	●	●	○	○	○	○	○	○	○	○	●	●	●	●
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Figure: Each news announcement comes with three sentiment scores $w_{positive} + w_{neutral} + w_{negative} = 1$, along with other scores for company relevance, novelty, etc.

On a given day, on average, about 1/3 of the S&P500 names have one (or more) relevant news announcements.

Performance of originally available news sentiment

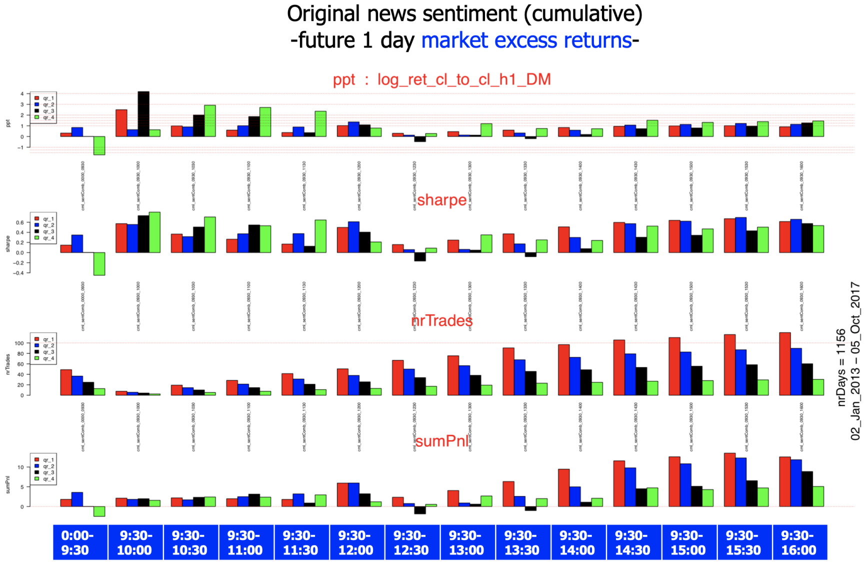
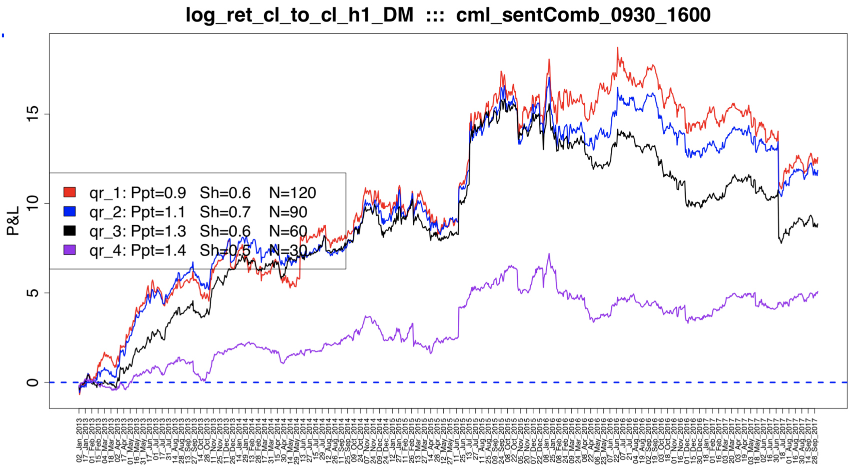


Figure: Performance statistics of quantile portfolios by time of day.

Cumulative P&L of originally available news sentiment

- Colors denote quantile portfolios.

Original news sentiment (cumulative) : 9:30 –16:00
-future 1 day market excess returns-



Knowledge graph Thomson Reuters

- supplierOf (2/3), hasCompetitor (1/3), isAffiliatedWith, hasUltimateParent
- supplier network is biggest graph
- competitor companies form clusters based on the industry
- affiliate network in S&P 500: investment and fund operators have the most connections

- How does news sentiment propagate through the graph? Are certain relations affected more than others?
- One way of assessing value from the graph information is by establishing a prediction task which could depend on the graph information.
- Understanding the effect of news on the companies for which the which is relevant, as well as the rest.

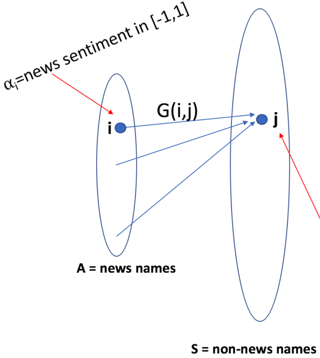
Knowledge graph Thomson Reuters

- supplierOf (2/3), hasCompetitor (1/3), isAffiliatedWith, hasUltimateParent
- supplier network is biggest graph
- competitor companies form clusters based on the industry
- affiliate network in S&P 500: investment and fund operators have the most connections

- How does news sentiment propagate through the graph? Are certain relations affected more than others?
- One way of assessing value from the graph information is by establishing a prediction task which could depend on the graph information.
- Understanding the effect of news on the companies for which the which is relevant, as well as the rest.

For now, build our own (correlation) network, and seek to propagate the available news sentiment on the network.

Propagation of Thomson Reuters news sentiment in a network



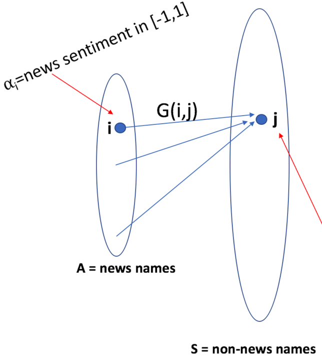
G(i,j) = C(i,j)*K(i,j)

C(i,j) = historical daily correlation

K(i,j) = knowledge graph
- -complete
- -supply
- -competitor

$$\alpha_j = \sum_i \alpha_i \, G(i,j)$$

Propagation of Thomson Reuters news sentiment in a network



G(i,j) = C(i,j)*K(i,j)

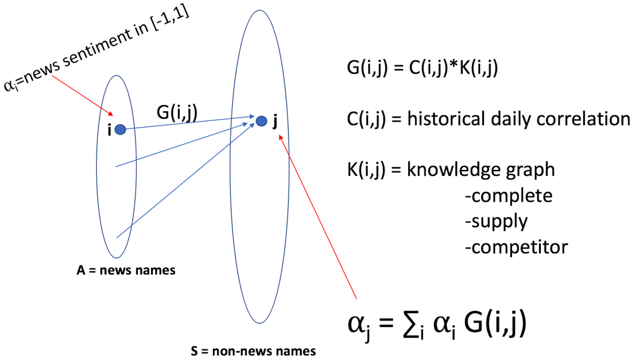
C(i,j) = historical daily correlation

K(i,j) = knowledge graph
- -complete
- -supply
- -competitor

$$\alpha_j = \sum_i \alpha_i \, G(i,j)$$

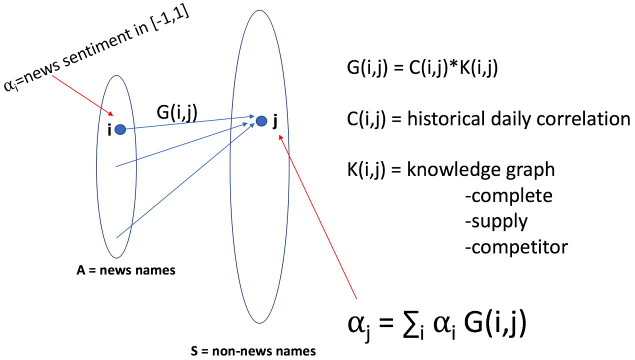
- explore interplay of performance with metrics based on {market cap, volatility, sector membership, centrality measures, past returns}

Propagation of Thomson Reuters news sentiment in a network



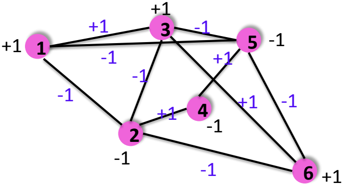
- explore interplay of performance with metrics based on {market cap, volatility, sector membership, centrality measures, past returns}
- evaluate news sentiment performance in light of low-rank structure of the market (systematic risk + idiosyncratic risk decomposition)

Propagation of Thomson Reuters news sentiment in a network



- explore interplay of performance with metrics based on {market cap, volatility, sector membership, centrality measures, past returns}
- evaluate news sentiment performance in light of low-rank structure of the market (systematic risk + idiosyncratic risk decomposition)
- interplay news/earnings dissemination and insider trading (correlate unusual trading activity with such events)

Synchronization over $\mathbb{Z}_2$



Synchronization over $\mathbb{Z}_2$

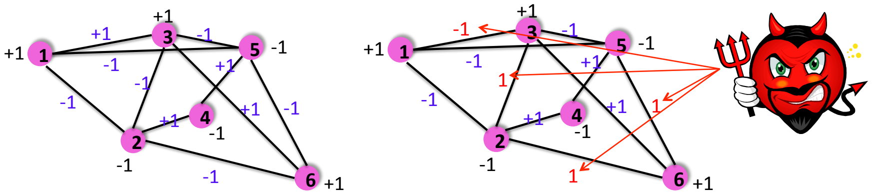


Figure: Left: clean. Right: noisy.

$$Z_{ij} = \begin{cases} z_i z_j^{-1} & (i,j) \in E; \text{clean}, \\ -z_i z_j^{-1} & (i,j) \in E; \text{corrupt}, \\ 0 & (i,j) \notin E \end{cases} \qquad \text{approx. solution} : x_1, \ldots, x_N$$

$$\max_{x_1,\ldots,x_N \in \mathbb{Z}_2^N} \sum_{i,j=1}^{N} x_i Z_{ij} x_j = \max_{x_1,\ldots,x_N \in \mathbb{Z}_2^N} x^T Z x, \quad \text{NP-hard}$$

- relax $\max_{\sum_{i=1}^N |x_i|^2 = N} \sum_{i,j=1}^N x_i Z_{ij} x_j = \max_{\|x\|^2 = N} x^T Z x$
- solved by the normalized top eigenvector $Z v_1 = \lambda_1 v_1$

Synchronization over $\mathbb{Z}_2$ with anchor information

Anchor set $\mathcal{A}$: group elements $a_i \in \mathbb{Z}_2$ that are known a-priori (given news)

Synchronization over $\mathbb{Z}_2$ with anchor information

Anchor set $\mathcal{A}$: group elements $a_i \in \mathbb{Z}_2$ that are known a-priori (given news)

Sensor set $\mathcal{S}$: group elements $s_i \in \mathbb{Z}_2$ to be estimated

Synchronization over $\mathbb{Z}_2$ with anchor information

Anchor set $\mathcal{A}$: group elements $a_i \in \mathbb{Z}_2$ that are known a-priori (given news)

Sensor set $\mathcal{S}$: group elements $s_i \in \mathbb{Z}_2$ to be estimated

$$Z = \begin{bmatrix} S & U \\ U^T & V \end{bmatrix}, \qquad D = \begin{bmatrix} D_S & 0 \\ 0 & D_V \end{bmatrix}$$

Synchronization over $\mathbb{Z}_2$ with anchor information

Anchor set $\mathcal{A}$: group elements $a_i \in \mathbb{Z}_2$ that are known a-priori (given news)
Sensor set $\mathcal{S}$: group elements $s_i \in \mathbb{Z}_2$ to be estimated

$$Z = \begin{bmatrix} S & U \\ U^T & V \end{bmatrix}, \qquad D = \begin{bmatrix} D_S & 0 \\ 0 & D_V \end{bmatrix}$$

- Relax to a quadratically const. quadratic program QCQP:

$$\begin{aligned} \underset{z=(z_1,\ldots,z_l)}{\text{minimize}} \quad & z^T(D_S - S)z - 2z^T Ua \\ \text{subject to} \quad & z^T z = I \end{aligned}$$

Synchronization over $\mathbb{Z}_2$ with anchor information

Anchor set $\mathcal{A}$: group elements $a_i \in \mathbb{Z}_2$ that are known a-priori (given news)

Sensor set $\mathcal{S}$: group elements $s_i \in \mathbb{Z}_2$ to be estimated

$$Z = \begin{bmatrix} S & U \\ U^T & V \end{bmatrix}, \qquad D = \begin{bmatrix} D_S & 0 \\ 0 & D_V \end{bmatrix}$$

- Relax to a quadratically const. quadratic program QCQP:

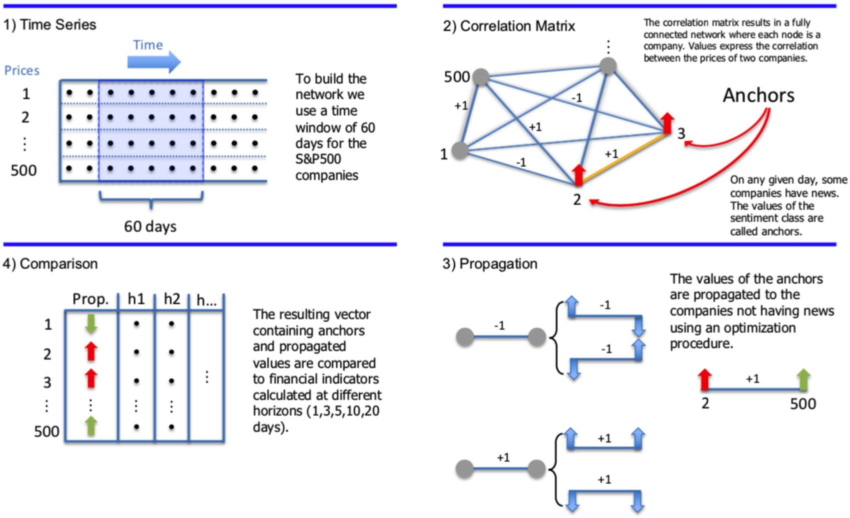
$$\begin{aligned} \underset{z=(z_1,\ldots,z_l)}{\text{minimize}} \quad & z^T(D_S - S)z - 2z^T Ua \\ \text{subject to} \quad & z^T z = I \end{aligned}$$

- SDP relaxation with anchors:

$$\begin{aligned} \underset{\Upsilon \in \mathbb{R}^{N\times N}}{\text{maximize}} \quad & Trace(Z\Upsilon) \\ \text{subject to} \quad & \Upsilon_{ii} = 1, i = 1, \ldots, N \\ & \Upsilon_{ij} = a_i a_j^{-1}, \quad \text{if } i, j \in \mathcal{A} \\ & \Upsilon \succeq 0 \end{aligned}$$

- Amenable to Burer-Monteiro approach

News propagation pipeline



Performance of propagated news sentiment (cumulative buckets)

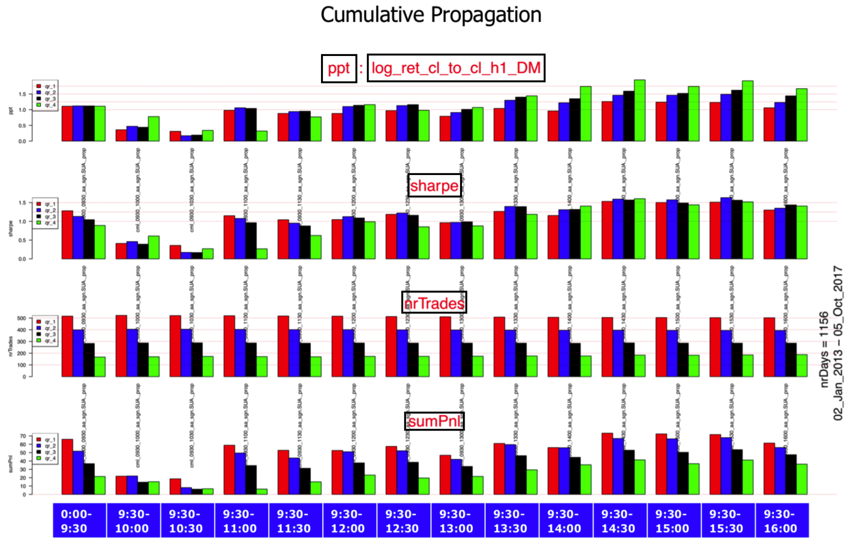
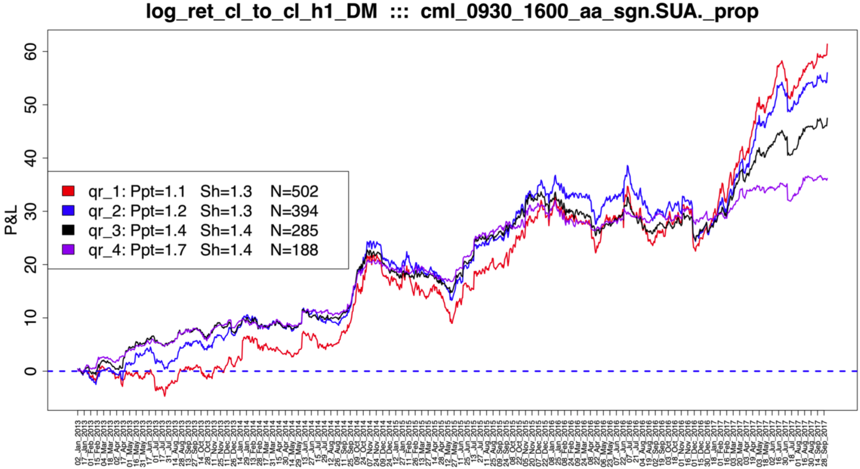


Figure: Performance statistics of quantile portfolios by time of day.

P&L of propagated news sentiment (cumulative buckets)

- Colors denote quantile portfolios.



qr_i: Quantile rank threshold
Ppt: P&L per trade in basis points
Sh: Sharpe Ratio
N: Number of instruments in the quantile portfolio

Performance of propagated news sentiment - individual buckets

(this includes the original news sentiments)

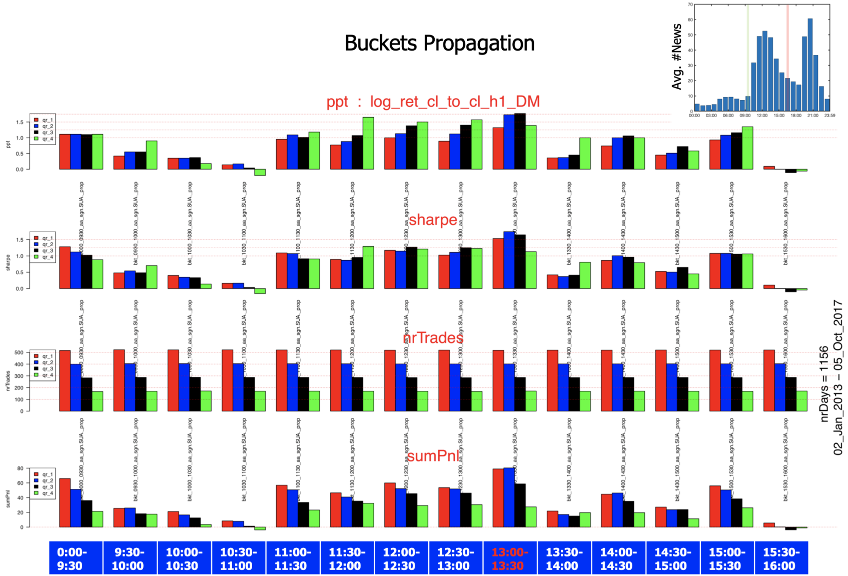
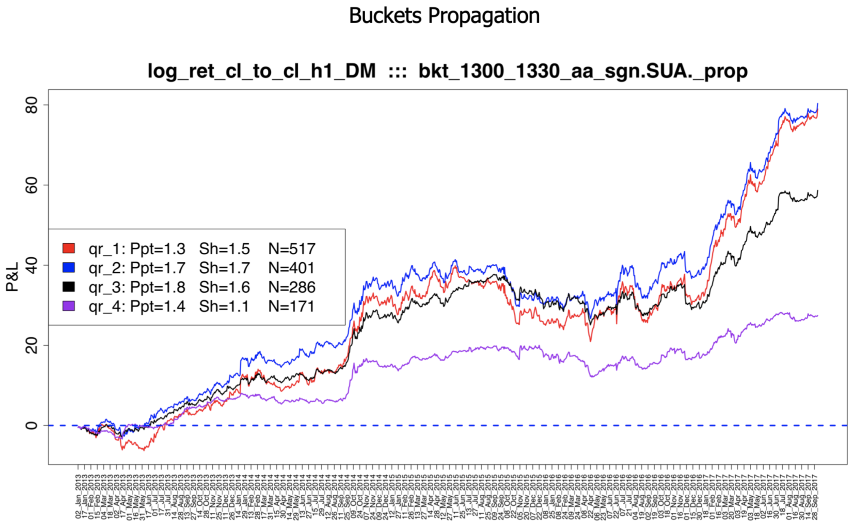


Figure: Performance statistics of quantile portfolios by time of day.

P&L of propagated news sentiment - 13:00-13:30 bucket

- Colors denote quantile portfolios.

Original news sentiments are included in the portfolio.



Signed Clustering & Signed Laplacians

- Kunegis et al. (2010): a unified approach to handle negative weights
  - inspiration from random walks, graph clustering, and electrical networks

Signed Clustering & Signed Laplacians

- Kunegis et al. (2010): a unified approach to handle negative weights
  - inspiration from random walks, graph clustering, and electrical networks
- solved a signed version of the 2-way ratio-cut problem via the Signed extension of the (combinatorial) graph Laplacian $\overline{L}$

$$\overline{L} = \overline{D} - A, \tag{7}$$

where $\overline{D}$ is a diagonal matrix with

$$\overline{D}_{ii} = \sum_{i=1}^{n} |d_{ij}|$$

Signed Clustering & Signed Laplacians

- Kunegis et al. (2010): a unified approach to handle negative weights
  - inspiration from random walks, graph clustering, and electrical networks
- solved a signed version of the 2-way ratio-cut problem via the Signed extension of the (combinatorial) graph Laplacian $\overline{L}$

$$\overline{L} = \overline{D} - A, \tag{7}$$

where $\overline{D}$ is a diagonal matrix with

$$\overline{D}_{ii} = \sum_{i=1}^{n} |d_{ij}|$$

- $\overline{L}$ is a positive-semidefinite matrix ($\lambda_i \geq 0, \forall i = 1, \ldots, n$), $\forall G$

Signed Clustering & Signed Laplacians

- Kunegis et al. (2010): a unified approach to handle negative weights
  - inspiration from random walks, graph clustering, and electrical networks
- solved a signed version of the 2-way ratio-cut problem via the Signed extension of the (combinatorial) graph Laplacian $\overline{L}$

$$\overline{L} = \overline{D} - A, \tag{7}$$

where $\overline{D}$ is a diagonal matrix with

$$\overline{D}_{ii} = \sum_{i=1}^{n} |d_{ij}|$$

- $\overline{L}$ is a positive-semidefinite matrix ($\lambda_i \geq 0, \forall i = 1, \ldots, n$), $\forall G$
- the random-walk normalized Laplacian

$$\overline{L}_{\text{rw}} = I - \overline{D}^{-1} A \tag{8}$$

Signed Clustering & Signed Laplacians

- Kunegis et al. (2010): a unified approach to handle negative weights
  - inspiration from random walks, graph clustering, and electrical networks
- solved a signed version of the 2-way ratio-cut problem via the Signed extension of the (combinatorial) graph Laplacian $\overline{L}$

$$\overline{L} = \overline{D} - A, \tag{7}$$

  where $\overline{D}$ is a diagonal matrix with

$$\overline{D}_{ii} = \sum_{i=1}^{n} |d_{ij}|$$

- $\overline{L}$ is a positive-semidefinite matrix ($\lambda_i \geq 0, \forall i = 1, \ldots, n$), $\forall G$
- the random-walk normalized Laplacian

$$\overline{L}_{\mathrm{rw}} = I - \overline{D}^{-1} A \tag{8}$$

- the symmetric graph Laplacian

$$\overline{L}_{\mathrm{sym}} = I - \overline{D}^{-1/2} A \overline{D}^{-1/2} \tag{9}$$

  (particularly suitable for skewed degree distributions)

• The top eigenvalues and eigenvectors of the Signed Laplacians contain information about the topological structure of the network.

• Knyazev (2018) argued the standard graph Laplacian is preferable for spectral clustering of signed graphs compared to $\overline{L}$; (we found no evidence)

Balanced Ratio/Normalized Cut

Kai-Yang Chiang, Joyce Whang, and Inderjit S. Dhillon. *Scalable Clustering of Signed Networks using Balance Normalized Cut, CIKM 2012*.

- Let $\{x_1, \ldots, x_k\} \in I$ denotes a k-cluster indicator set,
- Each of the k vectors is given by

$$x_t(i) = \begin{cases} 1 & \text{if node } i \in C_t \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Balanced Ratio/Normalized Cut

Kai-Yang Chiang, Joyce Whang, and Inderjit S. Dhillon. *Scalable Clustering of Signed Networks using Balance Normalized Cut, CIKM 2012*.

- Let $\{x_1, \ldots, x_k\} \in I$ denotes a k-cluster indicator set,
- Each of the k vectors is given by

$$x_t(i) = \begin{cases} 1 & \text{if node } i \in C_t \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

- Note the following algebraic manipulations

$$A = A^{+} - A^{-} \tag{11}$$

$$D^{+} - A = D^{+} - (A^{+} - A^{-})$$

Balanced Ratio/Normalized Cut

Kai-Yang Chiang, Joyce Whang, and Inderjit S. Dhillon. *Scalable Clustering of Signed Networks using Balance Normalized Cut, CIKM 2012*.

- Let $\{x_1, \ldots, x_k\} \in I$ denotes a k-cluster indicator set,
- Each of the k vectors is given by

$$x_t(i) = \begin{cases} 1 & \text{if node } i \in C_t \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

- Note the following algebraic manipulations

$$A = A^{+} - A^{-} \tag{11}$$

$$D^{+} - A = D^{+} - (A^{+} - A^{-}) = D^{+} - A^{+} + A^{-} = L^{+} + A^{-} \tag{12}$$

Balanced Ratio/Normalized Cut

Kai-Yang Chiang, Joyce Whang, and Inderjit S. Dhillon. *Scalable Clustering of Signed Networks using Balance Normalized Cut, CIKM 2012*.

- Let $\{x_1, \ldots, x_k\} \in I$ denotes a k-cluster indicator set,
- Each of the k vectors is given by

$$x_t(i) = \begin{cases} 1 & \text{if node } i \in C_t \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

- Note the following algebraic manipulations

$$A = A^+ - A^- \tag{11}$$

$$D^+ - A = D^+ - (A^+ - A^-) = D^+ - A^+ + A^- = L^+ + A^- \tag{12}$$

- What does $x^T L^+ x$ count?

Balanced Ratio/Normalized Cut

Kai-Yang Chiang, Joyce Whang, and Inderjit S. Dhillon. *Scalable Clustering of Signed Networks using Balance Normalized Cut, CIKM 2012*.

- Let $\{x_1, \ldots, x_k\} \in I$ denotes a k-cluster indicator set,
- Each of the k vectors is given by

$$x_t(i) = \begin{cases} 1 & \text{if node } i \in C_t \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

- Note the following algebraic manipulations

$$A = A^{+} - A^{-} \tag{11}$$

$$D^{+} - A = D^{+} - (A^{+} - A^{-}) = D^{+} - A^{+} + A^{-} = L^{+} + A^{-} \tag{12}$$

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

Balanced Ratio/Normalized Cut

Kai-Yang Chiang, Joyce Whang, and Inderjit S. Dhillon. *Scalable Clustering of Signed Networks using Balance Normalized Cut, CIKM 2012*.

- Let $\{x_1, \ldots, x_k\} \in I$ denotes a k-cluster indicator set,
- Each of the k vectors is given by

$$x_t(i) = \begin{cases} 1 & \text{if node } i \in C_t \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

- Note the following algebraic manipulations

$$A = A^+ - A^- \tag{11}$$

$$D^+ - A = D^+ - (A^+ - A^-) = D^+ - A^+ + A^- = L^+ + A^- \tag{12}$$

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

Balanced Ratio/Normalized Cut

Kai-Yang Chiang, Joyce Whang, and Inderjit S. Dhillon. *Scalable Clustering of Signed Networks using Balance Normalized Cut, CIKM 2012*.

- Let $\{x_1, \ldots, x_k\} \in I$ denotes a k-cluster indicator set,
- Each of the k vectors is given by

$$x_t(i) = \begin{cases} 1 & \text{if node } i \in C_t \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

- Note the following algebraic manipulations

$$A = A^+ - A^- \tag{11}$$

$$D^+ - A = D^+ - (A^+ - A^-) = D^+ - A^+ + A^- = L^+ + A^- \tag{12}$$

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

Balanced Ratio Cut

$$\min_{\{x_1,\ldots,x_k\}\in I} \left(\sum_{c=1}^{k} \frac{x_c^T (D^+ - A) x_c}{x_c^T x_c} \right), \tag{13}$$

References on signed clustering

- (Signed Laplacian $\overline{L}$) Jeroome Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto William De Luca, and Sahin Albayrak. *Spectral analysis of signed graphs for clustering, prediction and visualization*. SDM, 10:559–570, 2010.
- (Balanced Normalized Cut) Kai-Yang Chiang, Joyce Whang, and Inderjit S. Dhillon. *Scalable Clustering of Signed Networks using Balance Normalized Cut*. In ACM Conference on Information and Knowledge Management (CIKM), oct 2012.
- (SPONGE) M. Cucuringu, P. Davies, A. Glielmo, H. Tyagi, *SPONGE: A generalized eigenproblem for clustering signed networks*, AISTATS 2019

Also, see the *Related literature* section in the last reference above for a very succinct review of the signed clustering literature.

Detour: constrained clustering

- data similarity graph S
- domain knowledge is specified as a set of *soft*
  - must-link (ML) constraints graph Q_{ML}
  - cannot-link (CL) constraints graph Q_{CL}

Detour: constrained clustering

- data similarity graph S
- domain knowledge is specified as a set of *soft*
  - must-link (ML) constraints graph Q_{ML}
  - cannot-link (CL) constraints graph Q_{CL}

A generalized eigenproblem for constrained clustering (AISTATS 2016)

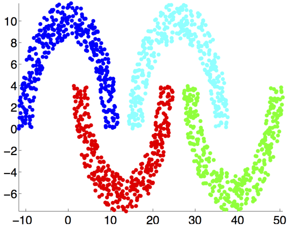
$$G = S + \lambda Q_{ML}$$
$$F = Q_{CL}$$
$$G\phi_i = \lambda_i F\phi_i, \; i = 1, \ldots, n$$

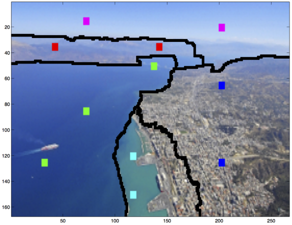
Detour: constrained clustering

- data similarity graph S
- domain knowledge is specified as a set of *soft*
  - must-link (ML) constraints graph Q_{ML}
  - cannot-link (CL) constraints graph Q_{CL}

A generalized eigenproblem for constrained clustering (AISTATS 2016)

$$G = S + \lambda Q_{ML}$$
$$F = Q_{CL}$$
$$G\phi_i = \lambda_i F \phi_i, \;\; i = 1, \dots, n$$





A generalized Cheeger inequality:

- lower bound on the minimum generalized eigenvalue of the pair of Laplacians L_G and L_F as function of the (generalized) conductance

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 $G = G^{+} \cup G^{-}$

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 $G = G^+ \cup G^-$

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

$$\frac{\mathrm{cut}_{G^+}(C, \overline{C})}{\mathrm{vol}_{G^+}(C)}, \tag{14}$$

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 $G = G^+ \cup G^-$

Constrained clustering ⇒ minimize the two measures of "*badness*"

$$\frac{\mathrm{cut}_{G^+}(C, \overline{C})}{\mathrm{vol}_{G^+}(C)}, \tag{14}$$

$$\left(\frac{\mathrm{cut}_{G^-}(C, \overline{C})}{\mathrm{vol}_{G^-}(C)}\right)^{-1} = \frac{\mathrm{vol}_{G^-}(C)}{\mathrm{cut}_{G^-}(C, \overline{C})}. \tag{15}$$

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 $G = G^+ \cup G^-$

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

$$\frac{\mathrm{cut}_{G^+}(C, \overline{C})}{\mathrm{vol}_{G^+}(C)}, \tag{14}$$

$$\left(\frac{\mathrm{cut}_{G^-}(C, \overline{C})}{\mathrm{vol}_{G^-}(C)}\right)^{-1} = \frac{\mathrm{vol}_{G^-}(C)}{\mathrm{cut}_{G^-}(C, \overline{C})}. \tag{15}$$

Ideally, want C s.t. both (14) and (15) are small. "Merge" obj. (14)+(15)

$$\min_{C \subset V} \frac{\mathrm{cut}_{G^+}(C, \overline{C}) + \tau^- \mathrm{vol}_{G^-}(C)}{\mathrm{cut}_{G^-}(C, \overline{C}) + \tau^+ \mathrm{vol}_{G^+}(C)}, \tag{16}$$

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

♦ Signed clustering: a generalized eigenproblem formulation

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

$$\min_{C_1,\cdots,C_k} \sum_{i=1}^{k} \frac{\mathrm{cut}_{G^+}(C_i, \overline{C_i}) + \tau^- \,\mathrm{vol}_{G^-}(C_i)}{\mathrm{cut}_{G^-}(C_i, \overline{C_i}) + \tau^+ \,\mathrm{vol}_{G^+}(C_i)}. \tag{17}$$

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

$$(x_{C_i})_j = \begin{cases} (\mathrm{cut}_{G^-}(C_i, \overline{C_i}) + \mathrm{vol}_{G^+}(C_i))^{-1/2}; & v_j \in C_i \\ 0; & v_j \notin C_i \end{cases} \tag{18}$$

renders (17) as the discrete optimization problem

$$\min_{C_1,\cdots,C_k} \sum_{i=1}^{k} \frac{x_{C_i}^T (L^+ + \tau^- D^-) x_{C_i}}{x_{C_i}^T (L^- + \tau^+ D^+) x_{C_i}}, \tag{19}$$

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 (19)

$$\min_{z_i (L^- + D^+) z_j = \delta_{ij}} \sum_{i=1}^{k} \frac{z_i^T (L^+ + \tau^- D^-) z_i}{z_i^T (L^- + \tau^+ D^+) z_i}. \tag{20}$$

- choice of $(L^- + \tau^+ D^+)$-orthonormality of vectors $z_1, \ldots, z_k$ is not a relaxation of (19); leads to a suitable eigenvalue problem
- assuming $L^- + \tau^+ D^+$ full rank, consider the change of variables

$$y_i = (L^- + \tau^+ D^+)^{1/2} z_i, \tag{21}$$

- changes the orthonormality constraints of (19) 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 (20) as

$$\min_{Y^T Y = I} \operatorname{Tr}\left(Y^T (L^- + \tau^+ D^+)^{-1/2} (L^+ + \tau^- D^-) (L^- + \tau^+ D^+)^{-1/2} Y \right) \tag{22}$$

◆ 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 (19)

$$\min_{z_i (L^- + D^+) z_j = \delta_{ij}} \sum_{i=1}^{k} \frac{z_i^T (L^+ + \tau^- D^-) z_i}{z_i^T (L^- + \tau^+ D^+) z_i}. \tag{20}$$

- choice of $(L^- + \tau^+ D^+)$-orthonormality of vectors $z_1, \ldots, z_k$ is not a relaxation of (19); leads to a suitable eigenvalue problem
- assuming $L^- + \tau^+ D^+$ full rank, consider the change of variables

$$y_i = (L^- + \tau^+ D^+)^{1/2} z_i, \tag{21}$$

- changes the orthonormality constraints of (19) 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 (20) as

$$\min_{Y^T Y = I} \mathrm{Tr} \left(Y^T (L^- + \tau^+ D^+)^{-1/2} (L^+ + \tau^- D^-) (L^- + \tau^+ D^+)^{-1/2} Y \right) \tag{22}$$

- solution: eigenvectors to the k-smallest eigenvalues of $T = (L^- + \tau^+ D^+)^{-1/2} (L^+ + \tau^- D^-) (L^- + \tau^+ D^+)^{-1/2}$

A generalized eigenvalue problem

The problem of finding a vector x that satisfied

$$Av = \lambda Bv \tag{23}$$

- λ is called a generalized eigenvalue (solves $\det(A - \lambda B) = 0$)
- x is called a generalized eigenvector

Algorithm 1 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++

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++

SPONGE$_{sym}$: uses the smallest k generalized eigenvectors of
$$(L^+_{sym} + \tau^- I, L^-_{sym} + \tau^+ I)$$

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}).

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

$\underline{i, j \text{ lie in } \textbf{same} \text{ cluster}}$

$$A_{ij} = \begin{cases} 1 & ; \quad \text{w. p} \quad p(1-\eta) \quad \text{correct} \\ -1 & ; \quad \text{w. p} \quad p\eta \quad \text{noisy} \\ 0 & ; \quad \text{w. p} \quad (1-p) \quad \text{missing} \end{cases} \tag{24}$$

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

$$A_{ij} = \begin{cases} 1 & ; \text{ w. p } & p(1-\eta) & \text{correct} \\ -1 & ; \text{ w. p } & p\eta & \text{noisy} \\ 0 & ; \text{ w. p } & (1-p) & \text{missing} \end{cases} \tag{24}$$

i, j lie in **different** clusters

$$A_{ij} = \begin{cases} 1 & ; \text{ w. p } & p\eta & \text{noisy} \\ -1 & ; \text{ w. p } & p(1-\eta) & \text{correct} \\ 0 & ; \text{ w. p } & (1-p) & \text{missing} \end{cases} . \tag{25}$$

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

<u>i, j lie in **same** cluster</u>

$$A_{ij} = \begin{cases} 1 & ; \text{ w. p } \quad p(1-\eta) \quad \text{correct} \\ -1 & ; \text{ w. p } \quad p\eta \quad \text{noisy} \\ 0 & ; \text{ w. p } \quad (1-p) \quad \text{missing} \end{cases} \tag{24}$$

<u>i, j lie in **different** clusters</u>

$$A_{ij} = \begin{cases} 1 & ; \text{ w. p } \quad p\eta \quad \text{noisy} \\ -1 & ; \text{ w. p } \quad p(1-\eta) \quad \text{correct} \\ 0 & ; \text{ w. p } \quad (1-p) \quad \text{missing} \end{cases} . \tag{25}$$

- performance guarantees for SPONGE for SSBM, for $k = 2$

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

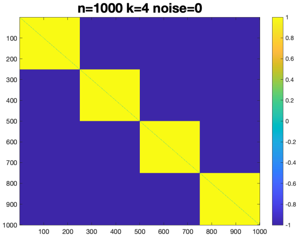
$$A_{ij} = \begin{cases} 1 & ; \quad \text{w. p} \quad p(1-\eta) \quad \text{correct} \\ -1 & ; \quad \text{w. p} \quad p\eta \quad \text{noisy} \\ 0 & ; \quad \text{w. p} \quad (1-p) \quad \text{missing} \end{cases} \tag{24}$$

i, j lie in **different** clusters

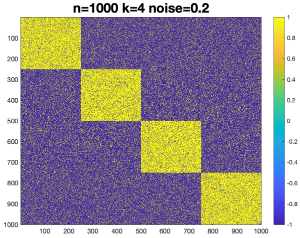
$$A_{ij} = \begin{cases} 1 & ; \quad \text{w. p} \quad p\eta \quad \text{noisy} \\ -1 & ; \quad \text{w. p} \quad p(1-\eta) \quad \text{correct} \\ 0 & ; \quad \text{w. p} \quad (1-p) \quad \text{missing} \end{cases} . \tag{25}$$

- performance guarantees for SPONGE for SSBM, for $k = 2$
- $\mathcal{R}(V_2(T))$ is close to $\mathcal{R}(V_2(\overline{T}))$ with high probability, provided n, p are large enough.

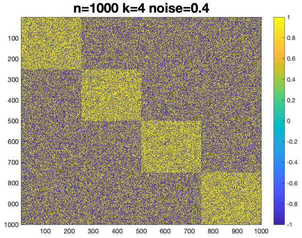
SSBM instances at varying noise levels



(a) $\eta = 0$



(b) $\eta = 0.2$



(c) $\eta = 0.4$

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

Theoretical 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

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

Theoretical 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

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

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

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

Rand Index

- $V = (v_1, v_2, \ldots, v_n)$ denotes set of vertices we wish to cluster/partition

♣ Rand Index

- $V = (v_1, v_2, \ldots, v_n)$ denotes set of vertices we wish to cluster/partition
- consider two partitions of V
  - $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ $\qquad$ $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_s\}$

♣ Rand Index

- $V = (v_1, v_2, \ldots, v_n)$ denotes set of vertices we wish to cluster/partition
- consider two partitions of V
  - $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_s\}$
- let $\{a, b, c, d\}$ count the number of vertex pairs that are placed in the
  - a: same subset of $\mathcal{B}$ and in the same subset of $\mathcal{C}$

♣ Rand Index

- $V = (v_1, v_2, \ldots, v_n)$ denotes set of vertices we wish to cluster/partition
- consider two partitions of V
  - $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ $\quad\quad$ $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_s\}$
- let $\{a, b, c, d\}$ count the number of vertex pairs that are placed in the
  - a: same subset of $\mathcal{B}$ and in the same subset of $\mathcal{C}$
  - b: same subset of $\mathcal{B}$ but in different subsets of $\mathcal{C}$

♣ Rand Index

- $V = (v_1, v_2, \ldots, v_n)$ denotes set of vertices we wish to cluster/partition
- consider two partitions of V
  - $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ $\qquad$ $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_s\}$
- let $\{a, b, c, d\}$ count the number of vertex pairs that are placed in the
  - a: same subset of $\mathcal{B}$ and in the same subset of $\mathcal{C}$
  - b: same subset of $\mathcal{B}$ but in different subsets of $\mathcal{C}$
  - c: different subsets of $\mathcal{B}$ but in the same set of $\mathcal{C}$

♣ Rand Index

- $V = (v_1, v_2, \ldots, v_n)$ denotes set of vertices we wish to cluster/partition
- consider two partitions of V
  - $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ $\qquad \mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_s\}$
- let $\{a, b, c, d\}$ count the number of vertex pairs that are placed in the
  - a: same subset of $\mathcal{B}$ and in the same subset of $\mathcal{C}$
  - b: same subset of $\mathcal{B}$ but in different subsets of $\mathcal{C}$
  - c: different subsets of $\mathcal{B}$ but in the same set of $\mathcal{C}$
  - d: different subsets of $\mathcal{B}$ and in different subsets of $\mathcal{C}$

♣ Rand Index

- $V = (v_1, v_2, \ldots, v_n)$ denotes set of vertices we wish to cluster/partition
- consider two partitions of V
  - $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_s\}$
- let $\{a, b, c, d\}$ count the number of vertex pairs that are placed in the
  - a: same subset of $\mathcal{B}$ and in the same subset of $\mathcal{C}$
  - b: same subset of $\mathcal{B}$ but in different subsets of $\mathcal{C}$
  - c: different subsets of $\mathcal{B}$ but in the same set of $\mathcal{C}$
  - d: different subsets of $\mathcal{B}$ and in different subsets of $\mathcal{C}$
- the Rand Index R is given by

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \tag{26}$$

♣ Rand Index

- $V = (v_1, v_2, \ldots, v_n)$ denotes set of vertices we wish to cluster/partition
- consider two partitions of V
  - $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ $\quad\quad \mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_s\}$
- let $\{a, b, c, d\}$ count the number of vertex pairs that are placed in the
  - a: same subset of $\mathcal{B}$ and in the same subset of $\mathcal{C}$
  - b: same subset of $\mathcal{B}$ but in different subsets of $\mathcal{C}$
  - c: different subsets of $\mathcal{B}$ but in the same set of $\mathcal{C}$
  - d: different subsets of $\mathcal{B}$ and in different subsets of $\mathcal{C}$
- the Rand Index R is given by

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \tag{26}$$

- a+b is the number of agreements between $\mathcal{B}$ and $\mathcal{C}$

♣ Rand Index

- $V = (v_1, v_2, \dots, v_n)$ denotes set of vertices we wish to cluster/partition
- consider two partitions of V
  - $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_r\}$ $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_s\}$
- let $\{a, b, c, d\}$ count the number of vertex pairs that are placed in the
  - a: same subset of $\mathcal{B}$ and in the same subset of $\mathcal{C}$
  - b: same subset of $\mathcal{B}$ but in different subsets of $\mathcal{C}$
  - c: different subsets of $\mathcal{B}$ but in the same set of $\mathcal{C}$
  - d: different subsets of $\mathcal{B}$ and in different subsets of $\mathcal{C}$
- the Rand Index R is given by

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \tag{26}$$

- a+b is the number of agreements between $\mathcal{B}$ and $\mathcal{C}$
- c + d is the number of disagreements between $\mathcal{B}$ and $\mathcal{C}$

♣ Rand Index

- $V = (v_1, v_2, \ldots, v_n)$ denotes set of vertices we wish to cluster/partition
- consider two partitions of V
  - $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ $\quad\quad$ $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_s\}$
- let $\{a, b, c, d\}$ count the number of vertex pairs that are placed in the
  - a: same subset of $\mathcal{B}$ and in the same subset of $\mathcal{C}$
  - b: same subset of $\mathcal{B}$ but in different subsets of $\mathcal{C}$
  - c: different subsets of $\mathcal{B}$ but in the same set of $\mathcal{C}$
  - d: different subsets of $\mathcal{B}$ and in different subsets of $\mathcal{C}$
- the Rand Index R is given by

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \tag{26}$$

- a+b is the number of agreements between $\mathcal{B}$ and $\mathcal{C}$
- c + d is the number of disagreements between $\mathcal{B}$ and $\mathcal{C}$
- R gives the frequency of occurrence of agreements over the total pairs

♣ Rand Index

- $V = (v_1, v_2, \ldots, v_n)$ denotes set of vertices we wish to cluster/partition
- consider two partitions of V
  - $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ $\qquad \mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_s\}$
- let $\{a, b, c, d\}$ count the number of vertex pairs that are placed in the
  - a: same subset of $\mathcal{B}$ and in the same subset of $\mathcal{C}$
  - b: same subset of $\mathcal{B}$ but in different subsets of $\mathcal{C}$
  - c: different subsets of $\mathcal{B}$ but in the same set of $\mathcal{C}$
  - d: different subsets of $\mathcal{B}$ and in different subsets of $\mathcal{C}$
- the Rand Index R is given by

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \tag{26}$$

- a+b is the number of agreements between $\mathcal{B}$ and $\mathcal{C}$
- c + d is the number of disagreements between $\mathcal{B}$ and $\mathcal{C}$
- R gives the frequency of occurrence of agreements over the total pairs
- R is also the probability that $\mathcal{B}$ and $\mathcal{C}$ agree on a randomly chosen pair
- $R = 0$ means the clusterings do not agree on any pair of points
- $R = 1$ means the clusterings are exactly the same

♣ Rand Index

- $V = (v_1, v_2, \ldots, v_n)$ denotes set of vertices we wish to cluster/partition
- consider two partitions of V
  - $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r\}$ $\qquad \mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_s\}$
- let $\{a, b, c, d\}$ count the number of vertex pairs that are placed in the
  - a: same subset of $\mathcal{B}$ and in the same subset of $\mathcal{C}$
  - b: same subset of $\mathcal{B}$ but in different subsets of $\mathcal{C}$
  - c: different subsets of $\mathcal{B}$ but in the same set of $\mathcal{C}$
  - d: different subsets of $\mathcal{B}$ and in different subsets of $\mathcal{C}$
- the Rand Index R is given by

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \tag{26}$$

- a+b is the number of agreements between $\mathcal{B}$ and $\mathcal{C}$
- c + d is the number of disagreements between $\mathcal{B}$ and $\mathcal{C}$
- R gives the frequency of occurrence of agreements over the total pairs
- R is also the probability that $\mathcal{B}$ and $\mathcal{C}$ agree on a randomly chosen pair
- $R = 0$ means the clusterings do not agree on any pair of points
- $R = 1$ means the clusterings are exactly the same
- if we regard one of the partitions (say $\mathcal{C}$ as the ground truth), then R quantifies the percentage of correct decisions made by the algorithm

♣ Rand Index

- $V = (v_1, v_2, \dots, v_n)$ denotes set of vertices we wish to cluster/partition
- consider two partitions of V
  - $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_r\}$ $\quad\quad$ $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_s\}$
- let $\{a, b, c, d\}$ count the number of vertex pairs that are placed in the
  - a: same subset of $\mathcal{B}$ and in the same subset of $\mathcal{C}$
  - b: same subset of $\mathcal{B}$ but in different subsets of $\mathcal{C}$
  - c: different subsets of $\mathcal{B}$ but in the same set of $\mathcal{C}$
  - d: different subsets of $\mathcal{B}$ and in different subsets of $\mathcal{C}$
- the Rand Index R is given by

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \tag{26}$$

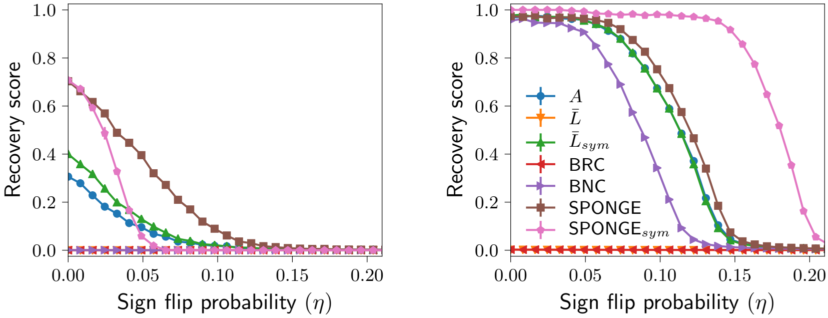
- a+b is the number of agreements between $\mathcal{B}$ and $\mathcal{C}$
- c + d is the number of disagreements between $\mathcal{B}$ and $\mathcal{C}$
- R gives the frequency of occurrence of agreements over the total pairs
- R is also the probability that $\mathcal{B}$ and $\mathcal{C}$ agree on a randomly chosen pair
- $R = 0$ means the clusterings do not agree on any pair of points
- $R = 1$ means the clusterings are exactly the same
- if we regard one of the partitions (say $\mathcal{C}$ as the ground truth), then R quantifies the percentage of correct decisions made by the algorithm

$$R = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{27}$$

- TF=true positives; TN=true negatives; FP=false positives; FN=false negatives.

Numerical comparison of state-of-the-art methods

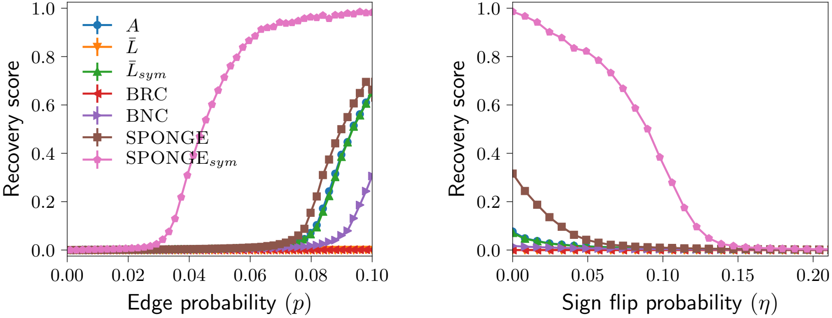
The recovery score used here is the Adjusted Rand Index (ARI) (a slightly modified version of the Rand Index).



(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.

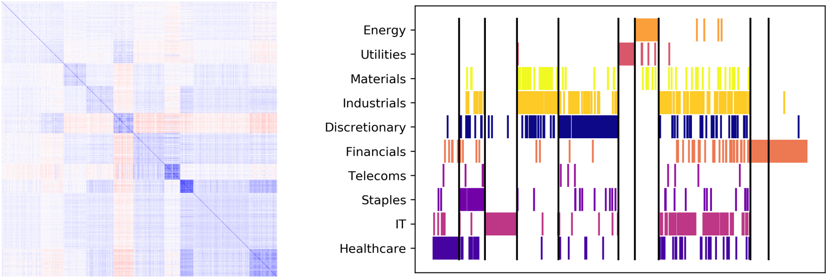
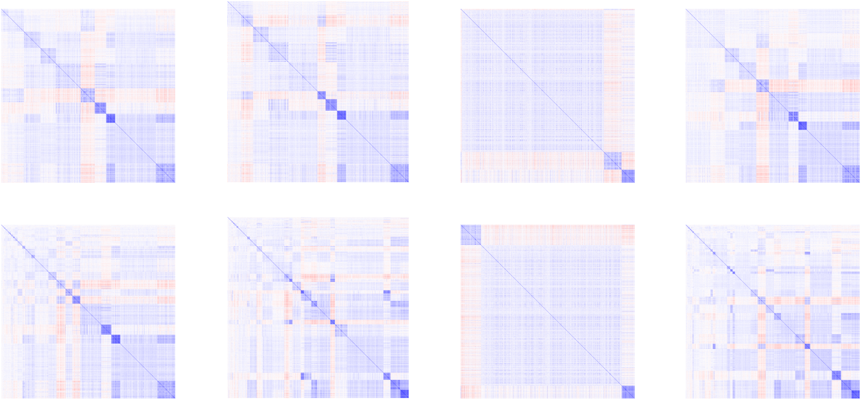


Figure: Left: the adjacency matrix A with rows/columns sorted in accordance to cluster membership. Right: Sector decomposition of the recovered clusters (based on a standard classification of the US economy into sectors).

Financial time series clustering

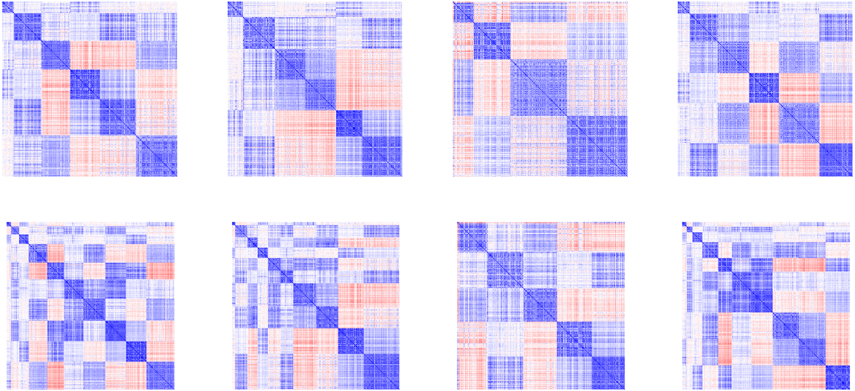


(e) SPONGE (f) SPONGE_{sym} (g) BNC (h) $\overline{L}_{sym}$

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

Australian rainfall data

- Clustering time series of historical rainfalls in $n = 306$ locations throughout Australia.



(e) SPONGE_{sym} (f) SPONGE_{sym} (g) BNC (h) $\overline{L}_{sym}$

Figure: Sorted adjacency matrix with $k = 6$ (top) and $k = 10$ (bottom).

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 $A_\tau = A + \tau \mathbf{1}\mathbf{1}^T$ ($\mathbf{1}$ 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 $A_\tau = A + \tau \mathbf{1}\mathbf{1}^T$ ($\mathbf{1}$ 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

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 , ≥ 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 , ≥ 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 , ≥ 0, define

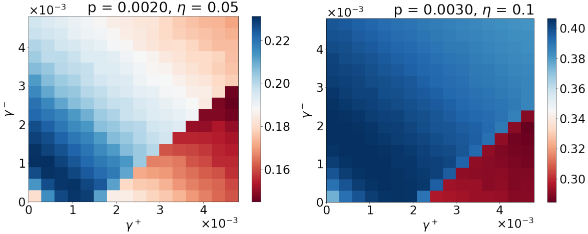
$$A^+_{\gamma^+} := A^+ + \frac{\gamma^+}{n}\mathbf{1}\mathbf{1}^\top; \qquad A^-_{\gamma^-} := A^- + \frac{\gamma^-}{n}\mathbf{1}\mathbf{1}^\top$$

  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 the regularized symmetric Signed Laplacian

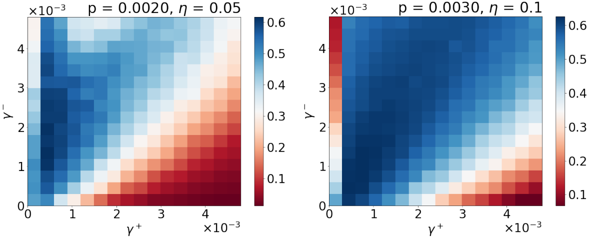
$$L_\gamma = I - (\overline{D}_\gamma)^{-1/2} A_\gamma (\overline{D}_\gamma)^{-1/2}, \tag{28}$$

  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



(b)

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

Polarization in signed graphs

- Han Xiao, Bruno Ordozgoiti, Aristides Gionis. 2020. *Searching for polarization in signed graphs: a local spectral approach.* (WWW 2020)

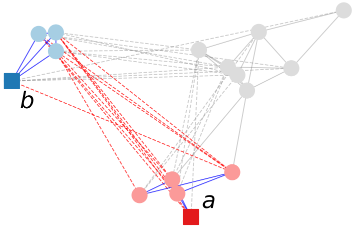
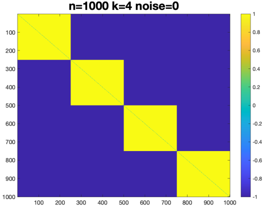


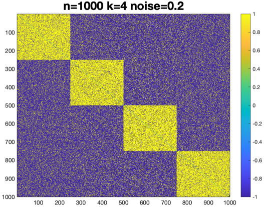
Figure: Example of polarization in a certain tribal network, with friendly relations shown as solid line and enemy relations as dashed line. Given a pair of node sets (S_1, S_2) as query, we are interested in finding two subgraphs C_1 and C_2 such that: (i) C_1 and C_2 are antagonistic to each other (ii) C_1 and $C2$ are friendly inside themselves and (iii) nodes in C_1 and C_2 are related to S_1 and S_2. Here $S_1 = \{a\}$, $S_2 = \{b\}$ as "seed nodes".

Polarization in signed graphs

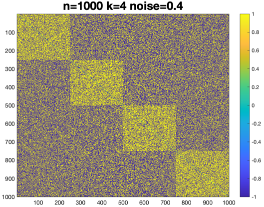
- Han Xiao, Bruno Ordozgoiti, Aristides Gionis. 2020. *Searching for polarization in signed graphs: a local spectral approach*. (WWW 2020)
- Kareem Darwish, *Quantifying Polarization on Twitter: the Kavanaugh Nomination*



(a) $\eta = 0$



(b) $\eta = 0.2$

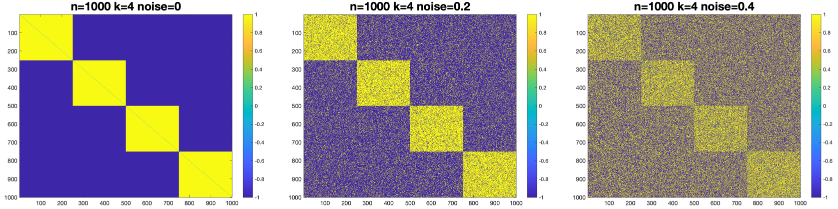


(c) $\eta = 0.4$

Figure: Standard SSBM with $n = 1000$, $k = 4$.

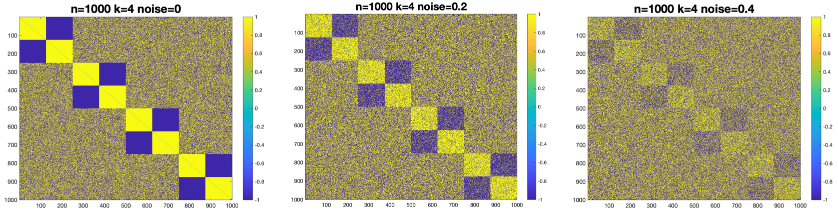
Polarization in signed graphs

- Han Xiao, Bruno Ordozgoiti, Aristides Gionis. 2020. *Searching for polarization in signed graphs: a local spectral approach*. (WWW 2020)
- Kareem Darwish, *Quantifying Polarization on Twitter: the Kavanaugh Nomination*



(a) $\eta = 0$ (b) $\eta = 0.2$ (c) $\eta = 0.4$

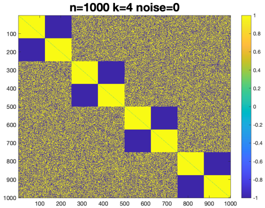
Figure: Standard SSBM with $n = 1000$, $k = 4$.



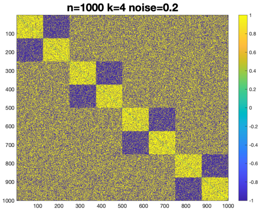
(a) $\eta = 0$ (b) $\eta = 0.2$ (c) $\eta = 0.4$

Figure: Polarized SSBM, $n = 1000$, $k = 4$.

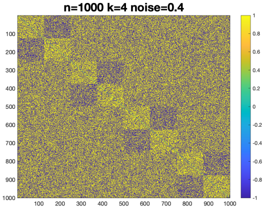
Planted polarized SSBM



(a) $\eta = 0$



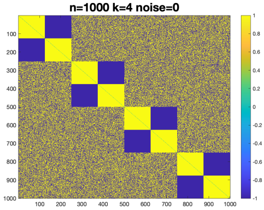
(b) $\eta = 0.2$



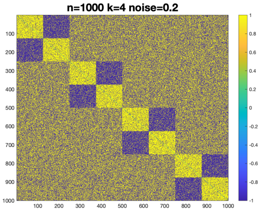
(c) $\eta = 0.4$

Figure: Polarized SSBM, $n = 1000$, $k = 4$.

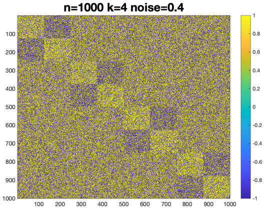
Planted polarized SSBM



(a) $\eta = 0$

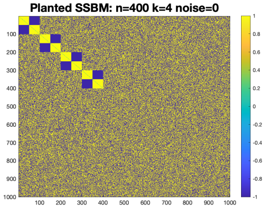


(b) $\eta = 0.2$

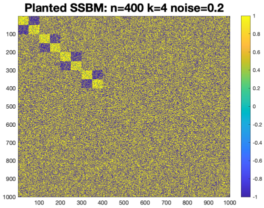


(c) $\eta = 0.4$

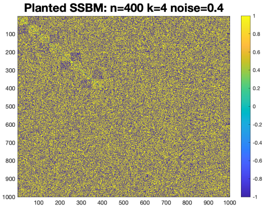
Figure: Polarized SSBM, $n = 1000$, $k = 4$.



(a) $\eta = 0$



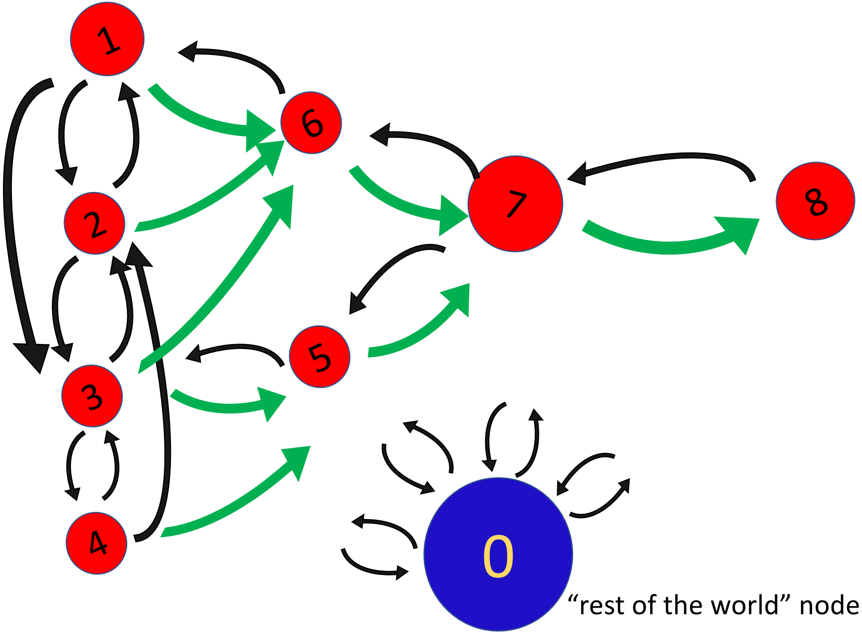
(b) $\eta = 0.2$



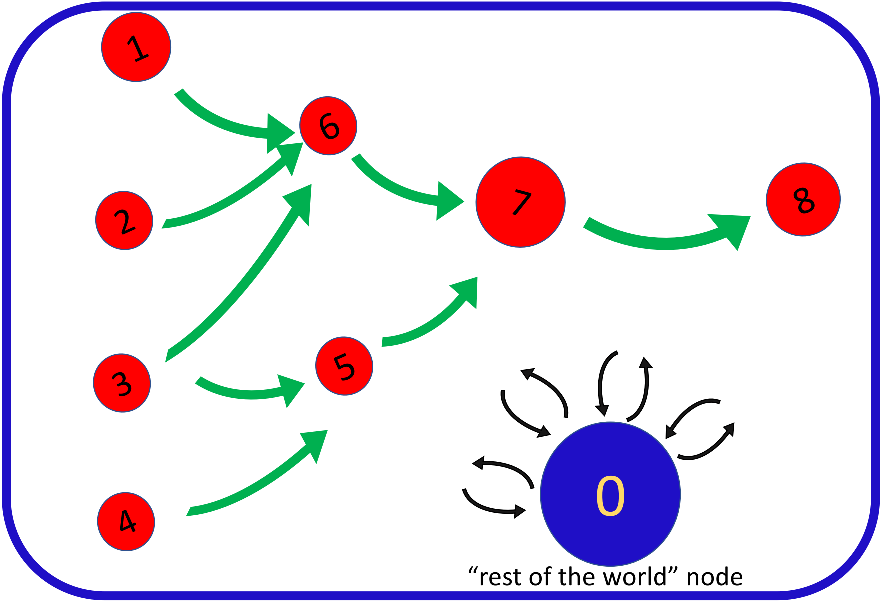
(c) $\eta = 0.4$

Figure: Planted polarized SSBM with $n = 400$, $k = 4$, and $m = 600$ ambient nodes.

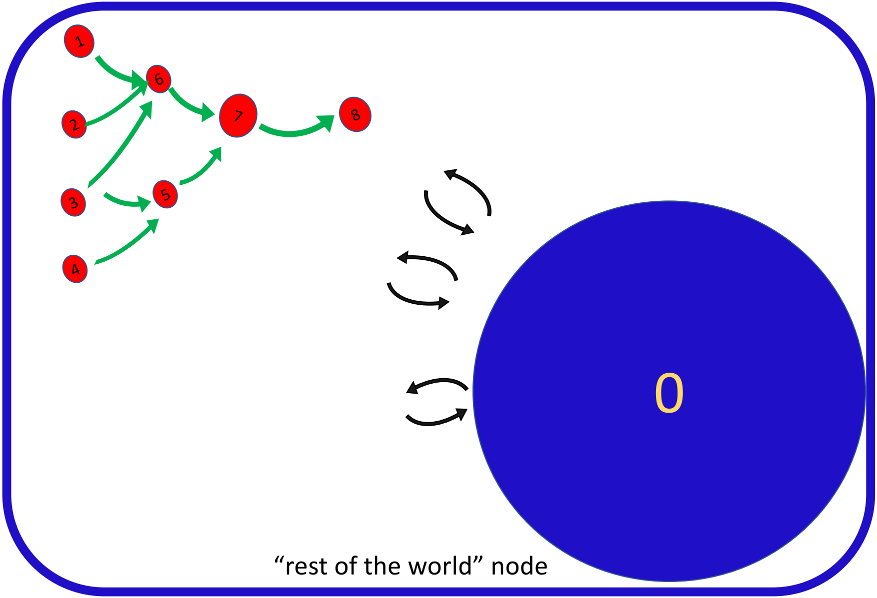
Transactions in a financial network (i)



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 $M \mapsto M + M^{\intercal}$

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 between counties in different states will be lost in the process

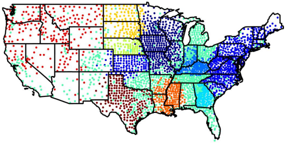
Directed graphs - challenges

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 between 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: $M + M^\top$

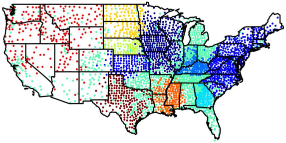
Directed graphs - challenges

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

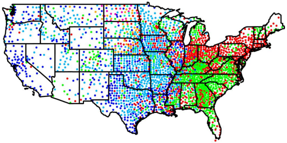
- usual normalized cut value (and similar clustering metrics based on edge-density) often fail to uncover many of the significant patterns
- asymmetric relationships contain essential structural information about the graph
- naive symmetrization of the adjacency matrix $M \mapsto M + M^\top$

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

- migration flows between 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

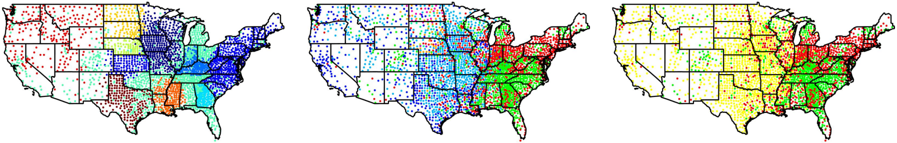
Directed graphs - challenges

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 between 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

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

- 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

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

- 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

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

- 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

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

- 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
- $p \in (0, 1]$ the probability there exists an edge between two vertices within the same 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
- $p \in (0, 1]$ the probability there exists an edge between two vertices within the same cluster
- $q \in [0, 1]$ the probability there exists an edge between two vertices belonging to two 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
- $p \in (0, 1]$ the probability there exists an edge between two vertices within the same cluster
- $q \in [0, 1]$ the probability there exists an edge between two vertices belonging to two different clusters
- matrix $F \in [0, 1]^{k \times k}$ controls the orientations of the edges between clusters and satisfies

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

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
- $p \in (0, 1]$ the probability there exists an edge between two vertices within the same cluster
- $q \in [0, 1]$ the probability there exists an edge between two vertices belonging to two different clusters
- matrix $F \in [0, 1]^{k \times k}$ controls the orientations of the edges between clusters and satisfies

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

- $\mathcal{G}\left(k, \{n_j\}_{j=0}^{k-1}, p, q, F\right)$

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
- $p \in (0, 1]$ the probability there exists an edge between two vertices within the same cluster
- $q \in [0, 1]$ the probability there exists an edge between two vertices belonging to two different clusters
- matrix $F \in [0, 1]^{k \times k}$ controls the orientations of the edges between clusters and satisfies

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

- $\mathcal{G}\left(k, \{n_j\}_{j=0}^{k-1}, p, q, F\right)$
- matrix F can be understood as the adjacency matrix of a weighted directed graph which represents the ***meta-graph*** describing the relations between the clusters.

Example

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

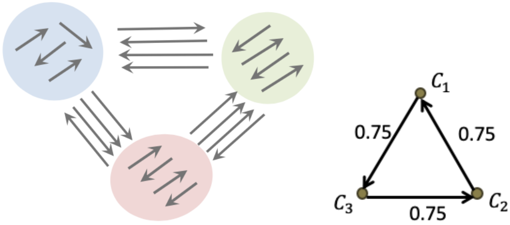


Figure: Circular flow.

- 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

Example

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

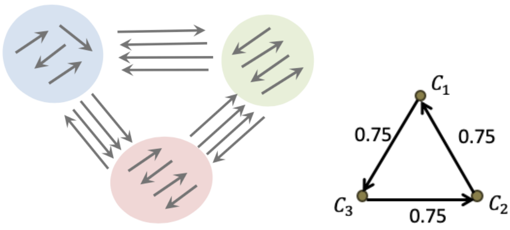


Figure: Circular flow.

- 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.

Example

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

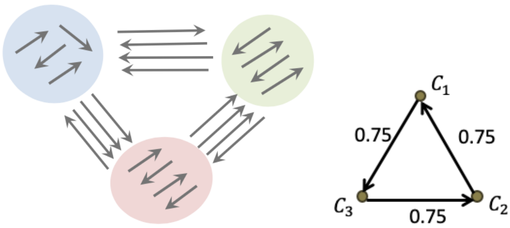


Figure: Circular flow.

- 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.

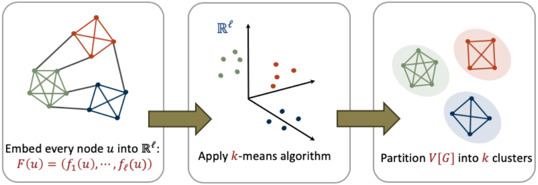
Algorithm 5 Spectral clustering for a digraph

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

1. Build Hermitian adjacency matrix A

$$A_{u,v} = \begin{cases} i & ; \text{ if } u \mapsto v \\ -i & ; \text{ if } v \mapsto u \\ 0 & ; \text{ if } \textit{otherwise} \end{cases} . \tag{29}$$

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
5. Return a partition of V corresponding to the output of k-means.



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

General F: number of misclassified vertices is $O\left(\frac{k^2 \log(kn)}{\tilde{\rho}^2 p}\right)$.

Why is this working so well?

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

$$(M^\top M)_{uv} = |\{w\colon w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}|, \tag{30}$$

$$(MM^\top)_{uv} = |\{w\colon u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|, \tag{31}$$

Why is this working so well?

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

$$(M^\top M)_{uv} = |\{w\colon w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}|, \tag{30}$$

$$(MM^\top)_{uv} = |\{w\colon u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|, \tag{31}$$

$$\begin{aligned}(M^\top M + MM^\top)_{uv} = {} & |\{w\colon w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}| \\ & + |\{w\colon u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|. \end{aligned} \tag{32}$$

Why is this working so well?

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

$$(M^\top M)_{uv} = |\{w\colon w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}|, \tag{30}$$

$$(MM^\top)_{uv} = |\{w\colon u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|, \tag{31}$$

$$\begin{aligned}(M^\top M + MM^\top)_{uv} = {} & |\{w\colon w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}| \\ & + |\{w\colon u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|.\end{aligned} \tag{32}$$

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

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

Why is this working so well?

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

$$(M^{\top}M)_{uv} = |\{w\colon w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}|, \tag{30}$$

$$(MM^{\top})_{uv} = |\{w\colon u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|, \tag{31}$$

$$(M^{\top}M + MM^{\top})_{uv} = |\{w\colon w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}| + |\{w\colon u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|. \tag{32}$$

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

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

- *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

Why is this working so well?

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

$$(M^\top M)_{uv} = |\{w\colon w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}|, \tag{30}$$

$$(MM^\top)_{uv} = |\{w\colon u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|, \tag{31}$$

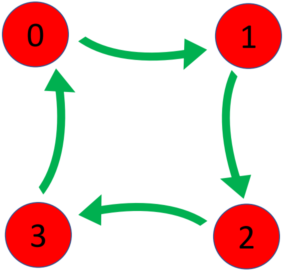
$$\begin{aligned}(M^\top M + MM^\top)_{uv} = {} & |\{w\colon w \rightsquigarrow u \text{ and } w \rightsquigarrow v\}| \\ & + |\{w\colon u \rightsquigarrow w \text{ and } v \rightsquigarrow w\}|. \end{aligned} \tag{32}$$

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

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

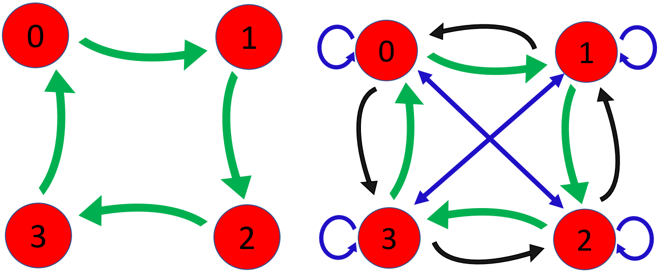
- A implicitly assigns a positive weight between a pair of vertices who have more common parents and offspring than ”mismatched" relations with third vertices, and a negative weight otherwise
- A implicitly keeps track of both common parents and offsprings without the need to perform an expensive matrix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

Meta-graph structures on k = 4 nodes



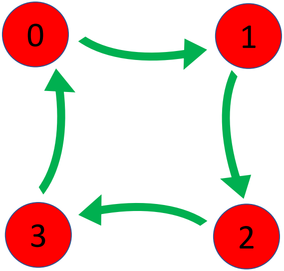
(a) Clean cycle meta-graph

Meta-graph structures on k = 4 nodes

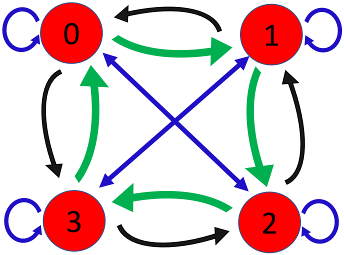


(a) Clean cycle meta-graph   (b) Noisy cycle meta-graph

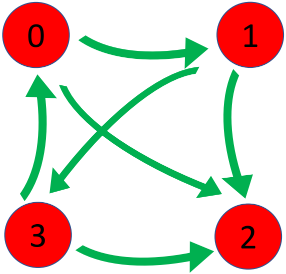
Meta-graph structures on $k = 4$ nodes



(a) Clean cycle meta-graph

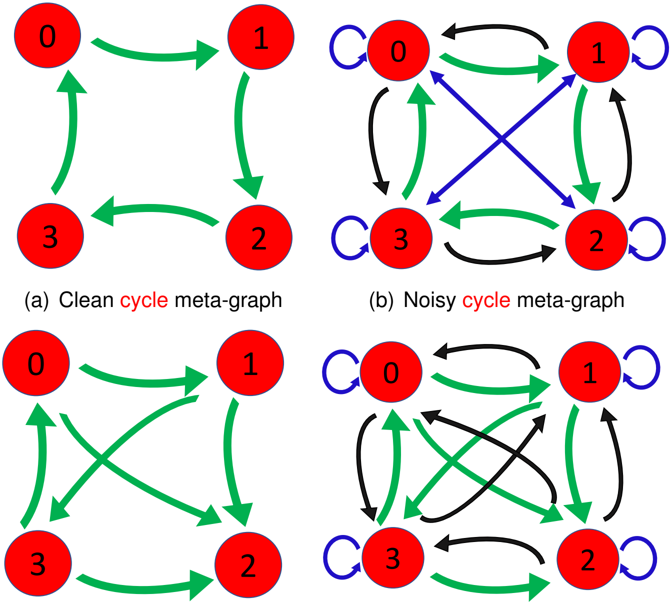


(b) Noisy cycle meta-graph



(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 $D_{jj} = \sum_{\ell=1}^{N} |A_{j\ell}|$

$$\textsc{Herm-RW}: \quad A_{rw} = D^{-1}A, \tag{33}$$

Normalization of A_G

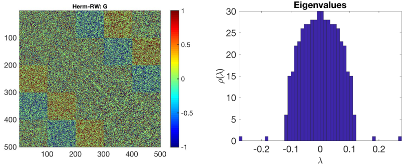
- D the diagonal matrix with $D_{jj} = \sum_{\ell=1}^{N} |A_{j\ell}|$

$$\text{HERM-RW}: \quad A_{rw} = D^{-1}A, \tag{33}$$

- similar to the Hermitian matrix

$$\text{HERM-SYM}: \quad A_{sym} = D^{-1/2}AD^{-1/2}, \tag{34}$$

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



(a) Circular pattern: G + Spectrum

Normalization of A_G

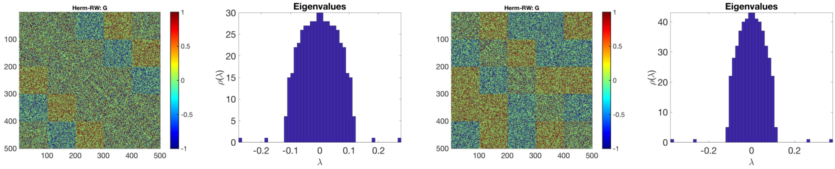
- D the diagonal matrix with $D_{jj} = \sum_{\ell=1}^{N} |A_{j\ell}|$

$$\text{HERM-RW}: \quad A_{rw} = D^{-1}A, \tag{33}$$

- similar to the Hermitian matrix

$$\text{HERM-SYM}: \quad A_{sym} = D^{-1/2}AD^{-1/2}, \tag{34}$$

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



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

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

Co-clustering directed graphs

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)

Co-clustering directed graphs

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)

Co-clustering the matrix yields two partitions of the same set of nodes

- the row clusters contain nodes with similar sending patterns
- the column clusters contain nodes with similar receiving patterns

Co-clustering directed graphs

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)

Co-clustering the matrix yields two partitions of the same set of nodes

- the row clusters contain nodes with similar sending patterns
- the column clusters contain nodes with similar receiving patterns
- to account for sparsity and heterogeneity in certain directed networks
  - uses a regularized graph Laplacian

$$L_{ij} = \frac{A_{ij}}{O^{\tau}_{ii} P^{\tau}_{jj}} = \left[(O^{\tau})^{-1/2} A (P^{\tau})^{-1/2} \right]_{ij} \qquad (35)$$

  - $P^{\tau}, O^{\tau} \in \mathbb{R}^{n \times n}$ diagonal matrices; $P^{\tau}_{jj} = \sum_k A_{kj} + \tau$; $O^{\tau}_{jj} = \sum_k A_{ik} + \tau$
  - regularization parameter set to the average out-degree $\tau = \sum_{i,k} A_{ik} / n$

Co-clustering directed graphs

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)

Co-clustering the matrix yields two partitions of the same set of nodes

- the row clusters contain nodes with similar sending patterns
- the column clusters contain nodes with similar receiving patterns
- to account for sparsity and heterogeneity in certain directed networks
  - uses a regularized graph Laplacian

$$L_{ij} = \frac{A_{ij}}{O^{\tau}_{ii} P^{\tau}_{jj}} = \left[(O^{\tau})^{-1/2} A (P^{\tau})^{-1/2} \right]_{ij} \tag{35}$$

  - $P^{\tau}, O^{\tau} \in \mathbb{R}^{n \times n}$ diagonal matrices; $P^{\tau}_{jj} = \sum_k A_{kj} + \tau$; $O^{\tau}_{jj} = \sum_k A_{ik} + \tau$
  - regularization parameter set to the average out-degree $\tau = \sum_{i,k} A_{ik} / n$
  - compute the top k left and right singular vectors $X_L \in \mathbb{R}^{n \times k}$ and $X_R \in \mathbb{R}^{n \times k}$

Co-clustering directed graphs

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)

Co-clustering the matrix yields two partitions of the same set of nodes

- the row clusters contain nodes with similar sending patterns
- the column clusters contain nodes with similar receiving patterns
- to account for sparsity and heterogeneity in certain directed networks
  - uses a regularized graph Laplacian

$$L_{ij} = \frac{A_{ij}}{O^\tau_{ii} P^\tau_{jj}} = \left[(O^\tau)^{-1/2} A (P^\tau)^{-1/2} \right]_{ij} \tag{35}$$

  - $P^\tau, O^\tau \in \mathbb{R}^{n \times n}$ diagonal matrices; $P^\tau_{jj} = \sum_k A_{kj} + \tau$; $O^\tau_{jj} = \sum_k A_{ik} + \tau$
  - regularization parameter set to the average out-degree $\tau = \sum_{i,k} A_{ik} / n$
  - compute the top k left and right singular vectors $X_L \in \mathbb{R}^{n \times k}$ and $X_R \in \mathbb{R}^{n \times k}$
  - projects the rows of the eigenvector matrix onto the sphere: normalize each row of X_L and X_R to have unit length

$$[X^*_L]_i = \frac{[X^*_L]_i}{||[X^*_L]_i||_2} \quad \text{and} \quad [X^*_R]_i = \frac{[X^*_R]_i}{||[X^*_R]_i||_2} \tag{36}$$

Co-clustering directed graphs

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)

Co-clustering the matrix yields two partitions of the same set of nodes

- the row clusters contain nodes with similar sending patterns
- the column clusters contain nodes with similar receiving patterns
- to account for sparsity and heterogeneity in certain directed networks
  - uses a regularized graph Laplacian

$$L_{ij} = \frac{A_{ij}}{O_{ii}^{\tau} P_{jj}^{\tau}} = \left[(O^{\tau})^{-1/2} A (P^{\tau})^{-1/2} \right]_{ij} \tag{35}$$

  - $P^{\tau}, O^{\tau} \in \mathbb{R}^{n \times n}$ diagonal matrices; $P_{jj}^{\tau} = \sum_k A_{kj} + \tau$; $O_{jj}^{\tau} = \sum_k A_{ik} + \tau$
  - regularization parameter set to the average out-degree $\tau = \sum_{i,k} A_{ik} / n$
  - compute the top k left and right singular vectors $X_L \in \mathbb{R}^{n \times k}$ and $X_R \in \mathbb{R}^{n \times k}$
  - projects the rows of the eigenvector matrix onto the sphere: normalize each row of X_L and X_R to have unit length

$$[X_L^*]_i = \frac{[X_L^*]_i}{||[X_L^*]_i||_2} \quad \text{and} \quad [X_R^*]_i = \frac{[X_R^*]_i}{||[X_R^*]_i||_2} \tag{36}$$

  - can either concatenate $[X_L^*]$ and $[X_R^*]$ and run k-means in the resulting embedding; or could run separately k-means on the rows of $[X_L^*]$ and $[X_R^*]$ (potentially with different number of clusters k_L and k_R)

Co-clustering directed graphs

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)

Co-clustering the matrix yields two partitions of the same set of nodes

- the row clusters contain nodes with similar sending patterns
- the column clusters contain nodes with similar receiving patterns
- to account for sparsity and heterogeneity in certain directed networks
  - uses a regularized graph Laplacian

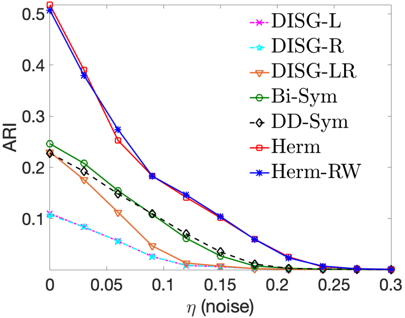
$$L_{ij} = \frac{A_{ij}}{O^{\tau}_{ii} P^{\tau}_{jj}} = \left[(O^{\tau})^{-1/2} A (P^{\tau})^{-1/2} \right]_{ij} \qquad (35)$$

  - $P^{\tau}, O^{\tau} \in \mathbb{R}^{n \times n}$ diagonal matrices; $P^{\tau}_{jj} = \sum_k A_{kj} + \tau$; $O^{\tau}_{jj} = \sum_k A_{ik} + \tau$
  - regularization parameter set to the average out-degree $\tau = \sum_{i,k} A_{ik} / n$
  - compute the top k left and right singular vectors $X_L \in \mathbb{R}^{n \times k}$ and $X_R \in \mathbb{R}^{n \times k}$
  - projects the rows of the eigenvector matrix onto the sphere: normalize each row of X_L and X_R to have unit length

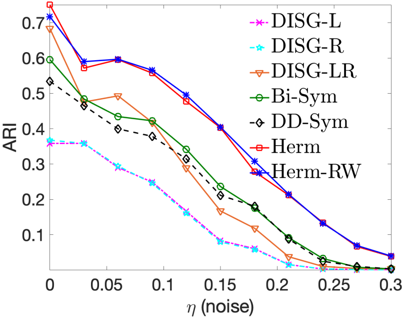
$$[X^*_L]_i = \frac{[X^*_L]_i}{||[X^*_L]_i||_2} \quad \text{and} \quad [X^*_R]_i = \frac{[X^*_R]_i}{||[X^*_R]_i||_2} \qquad (36)$$

  - can either concatenate $[X^*_L]$ and $[X^*_R]$ and run k-means in the resulting embedding; or could run separately k-means on the rows of $[X^*_L]$ and $[X^*_R]$ (potentially with different number of clusters k_L and k_R)
- advantage: A need not be skew symmetric

Comparison with state-of-the-art



(a) Circular pattern

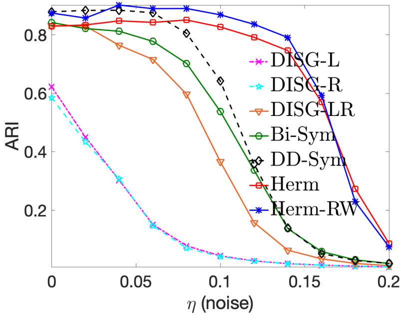


(b) Complete meta-graph

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)
- **{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*

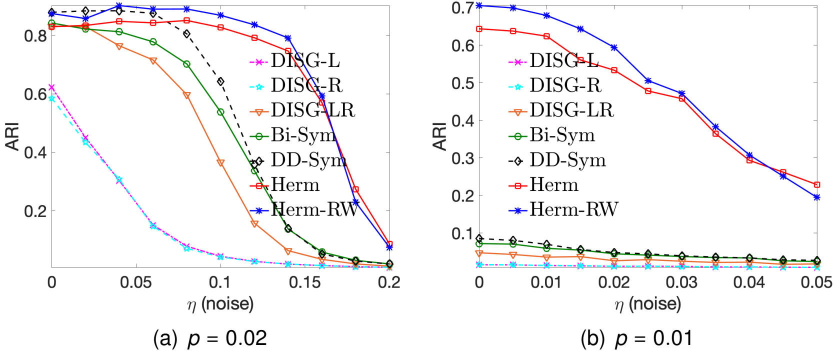
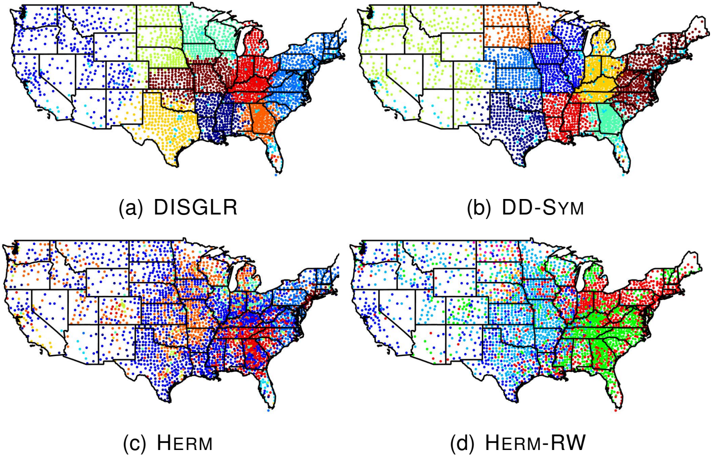


Figure: Recovery rates for the complete meta-graph in the DSBM with $k = 50$, $N = 5000$, two sparsity values p. Averaged over 10 runs.

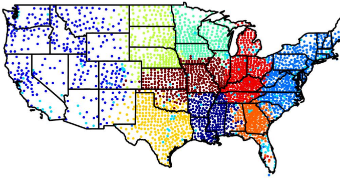
Clustering the US Migration Network



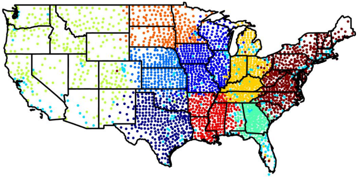
(a) DISGLR (b) DD-SYM

(c) HERM (d) HERM-RW

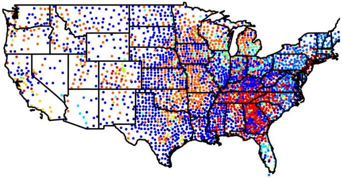
Clustering the US Migration Network



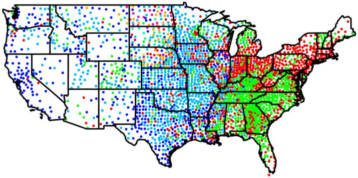
(e) DISGLR



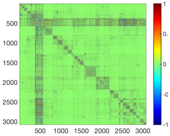
(f) DD-SYM



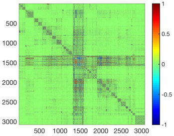
(g) HERM



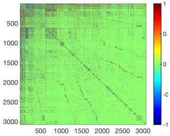
(h) HERM-RW



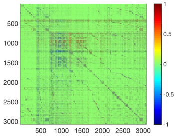
(a) DISGLR



(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

$$\mathrm{CI}(X, Y) = \frac{w(X, Y)}{w(X, Y) + w(Y, X)}, \tag{37}$$

- $w(X, Y) = \sum_{j \in X, \ell \in Y} w(j, \ell)$ 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

$$\mathrm{CI}(X, Y) = \frac{w(X, Y)}{w(X, Y) + w(Y, X)}, \tag{37}$$

- $w(X, Y) = \sum_{j \in X, \ell \in Y} w(j, \ell)$ 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

$$\mathrm{CI}^{\mathrm{size}}(X, Y) = |\mathrm{CI}(X, Y) - 0.5| \cdot \min\{|X|, |Y|\}, \tag{38}$$

Cut Imbalance Ratio

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

$$\mathrm{CI}(X, Y) = \frac{w(X, Y)}{w(X, Y) + w(Y, X)}, \tag{37}$$

- $w(X, Y) = \sum_{j \in X, \ell \in Y} w(j, \ell)$ 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

$$\mathrm{CI}^{\mathrm{size}}(X, Y) = |\mathrm{CI}(X, Y) - 0.5| \cdot \min\{|X|, |Y|\}, \tag{38}$$

$$\mathrm{CI}^{\mathrm{vol}}(X, Y) = |\mathrm{CI}(X, Y) - 0.5| \cdot \min\{\mathrm{vol}(X), \mathrm{vol}(Y)\}, \tag{39}$$

- $vol(X) = \sum_{j \in X} d_j^{in} + d_j^{out}$, 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

$$\mathrm{CI}(X, Y) = \frac{w(X, Y)}{w(X, Y) + w(Y, X)}, \tag{37}$$

- $w(X, Y) = \sum_{j \in X, \ell \in Y} w(j, \ell)$ 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

$$\mathrm{CI}^{\mathrm{size}}(X, Y) = |\mathrm{CI}(X, Y) - 0.5| \cdot \min\{|X|, |Y|\}, \tag{38}$$

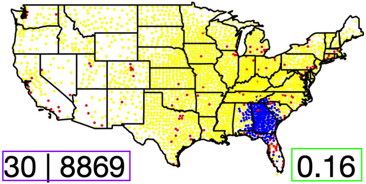
$$\mathrm{CI}^{\mathrm{vol}}(X, Y) = |\mathrm{CI}(X, Y) - 0.5| \cdot \min\{\mathrm{vol}(X), \mathrm{vol}(Y)\}, \tag{39}$$

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

$$\mathrm{TopCI}^{\mathrm{vol}} = \sum_{t=1}^{M} \mathrm{CI}^{\mathrm{vol}}(C_{j_t}, C_{\ell_t}) \tag{40}$$

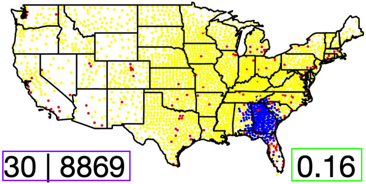
where (C_{j_t}, C_{ℓ_t}) denotes the t-th largest $\mathrm{CI}^{\mathrm{vol}}$ cut imbalance pair.

US Migration - top largest size-normalized cut imbalance pair

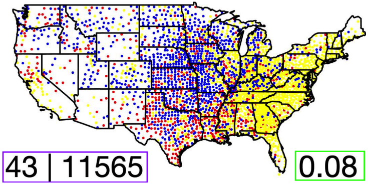


(a) DISGLR

US Migration - top largest size-normalized cut imbalance pair

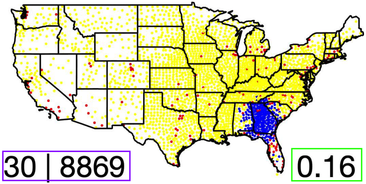


(a) DISGLR

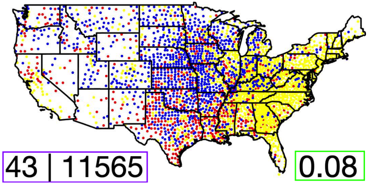


(b) HERM

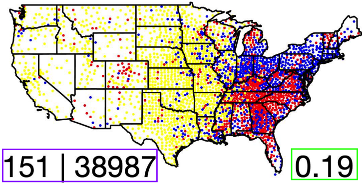
US Migration - top largest size-normalized cut imbalance pair



(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 $\mathrm{CI}^{\mathrm{size}}$ and $\mathrm{CI}^{\mathrm{vol}}$ 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).

Further topics within networks

- many networks have a hierarchical structure (food webs, GICS decomposition: sectors, groups, etc). How do we take this into account when, for example, denoising the empirical correlation matrix?

Further topics within networks

- many networks have a hierarchical structure (food webs, GICS decomposition: sectors, groups, etc). How do we take this into account when, for example, denoising the empirical correlation matrix?
- networks may also be dynamic; each given network data is a snapshot in time

Further topics within networks

- many networks have a hierarchical structure (food webs, GICS decomposition: sectors, groups, etc). How do we take this into account when, for example, denoising the empirical correlation matrix?
- networks may also be dynamic; each given network data is a snapshot in time
- edges in networks may be of different types (as in knowledge graphs), leading to multilayer/multiplex networks

Further topics within networks

- many networks have a hierarchical structure (food webs, GICS decomposition: sectors, groups, etc). How do we take this into account when, for example, denoising the empirical correlation matrix?
- networks may also be dynamic; each given network data is a snapshot in time
- edges in networks may be of different types (as in knowledge graphs), leading to multilayer/multiplex networks
- certain networks have structures beyond the usual clustering, eg core-periphery

Further topics within networks

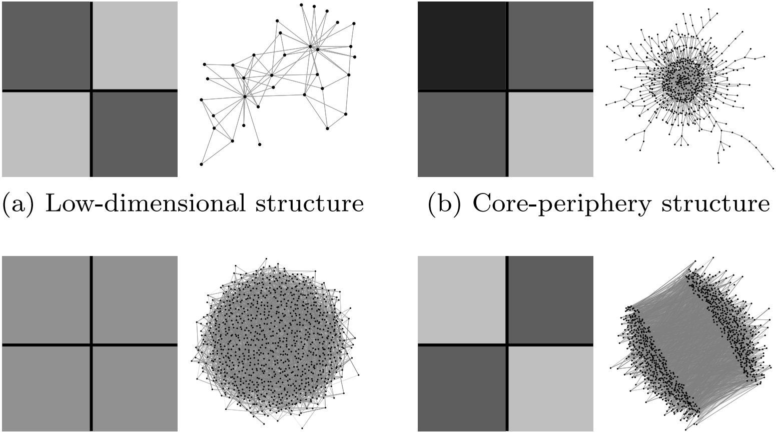
- many networks have a hierarchical structure (food webs, GICS decomposition: sectors, groups, etc). How do we take this into account when, for example, denoising the empirical correlation matrix?
- networks may also be dynamic; each given network data is a snapshot in time
- edges in networks may be of different types (as in knowledge graphs), leading to multilayer/multiplex networks
- certain networks have structures beyond the usual clustering, eg core-periphery
- anomaly detection (fraud detection in financial transaction networks)

Further topics within networks

- many networks have a hierarchical structure (food webs, GICS decomposition: sectors, groups, etc). How do we take this into account when, for example, denoising the empirical correlation matrix?
- networks may also be dynamic; each given network data is a snapshot in time
- edges in networks may be of different types (as in knowledge graphs), leading to multilayer/multiplex networks
- certain networks have structures beyond the usual clustering, eg core-periphery
- anomaly detection (fraud detection in financial transaction networks)
- network change-point detection (NCPD) (identification of market regimes)

A range of structures in networks

Idealized block models of network adjacency matrices; darker blocks correspond to denser connections among its component nodes.



(a) Low-dimensional structure

(b) Core-periphery structure

(c) Expander or complete graph

(d) Bipartite structure

Jeub, L. G., Balachandran, P., Porter, M. A., Mucha, P. J., & Mahoney, M. W. (2015). Think locally, act locally: Detection of small, medium-sized, and large communities in large networks. Physical Review E, 91(1), 012821

Temporal (or time-dependent) networks

- networks for which either the nodes or the edges (or both) change over time

Temporal (or time-dependent) networks

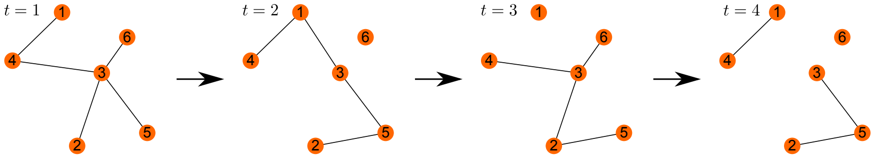
- networks for which either the nodes or the edges (or both) change over time
- voting networks (U.S. Senate roll call vote similarities) encoding the pairwise agreement/similarity between their voting, specified independently for each 2-year Congress

Temporal (or time-dependent) networks

- networks for which either the nodes or the edges (or both) change over time
- voting networks (U.S. Senate roll call vote similarities) encoding the pairwise agreement/similarity between their voting, specified independently for each 2-year Congress
- social networks where new connections are formed over time, and also new users join the network.

Temporal (or time-dependent) networks

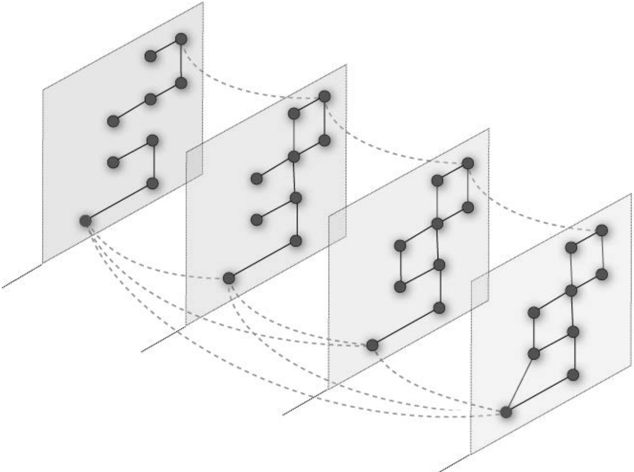
- networks for which either the nodes or the edges (or both) change over time
- voting networks (U.S. Senate roll call vote similarities) encoding the pairwise agreement/similarity between their voting, specified independently for each 2-year Congress
- social networks where new connections are formed over time, and also new users join the network.
- daily correlation matrices, co-occurrence network, co-jumps networks



Credit: Heather Harrington

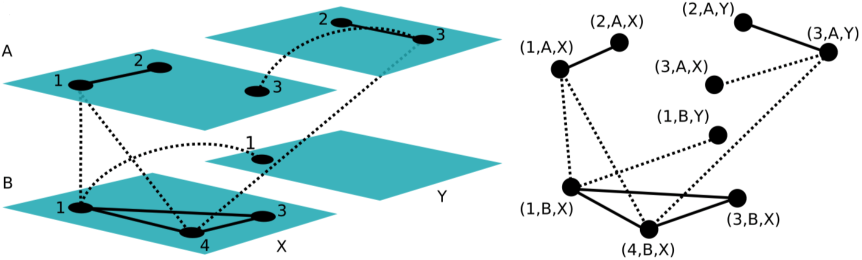
Temporal (or time-dependent) networks

- network for which either the nodes or the edges (or both) change over time
- *interslice connections* (dashed lines) are encoded by T_{jrs}, specifying the coupling of node j to itself between slices r and s

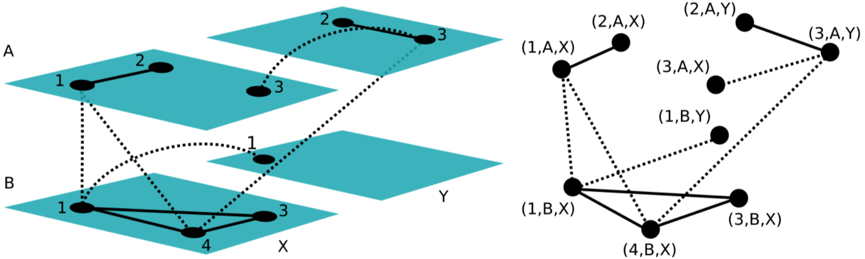


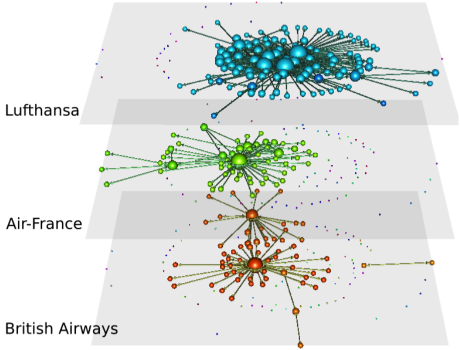
Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., & Onnela, J. P. *Community structure in time-dependent, multiscale, and multiplex networks*. Science, 328(5980), 876-878 (2010)

Multilayer networks



Multilayer networks





Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., & Porter, M. A. (2014). *Multilayer networks*. Journal of complex networks, 2(3), 203-271.

Hypergraphs

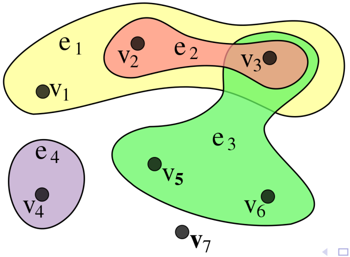
- certain applications require moving beyond pairwise connections
- eg., co-authorship networks: multiple authors write a joint paper

Hypergraphs

- certain applications require moving beyond pairwise connections
- eg., co-authorship networks: multiple authors write a joint paper
- in hypergraphs, the edges (denoted as hyperedges) are allowed to connect more than two nodes
- k-regular hypergraph: every node has degree k, i.e., it is contained exactly in k hyperedges
- often represented as *tensors*

Hypergraphs

- certain applications require moving beyond pairwise connections
- eg., co-authorship networks: multiple authors write a joint paper
- in hypergraphs, the edges (denoted as hyperedges) are allowed to connect more than two nodes
- *k*-regular hypergraph: every node has degree *k*, i.e., it is contained exactly in *k* hyperedges
- often represented as *tensors*
- in a financial context, it can be used to capture
  - co-jump behaviour: events in which a subset of stocks co-jump within the same short time interval
  - co-occurrence events: multiple companies mentioned in the same news article



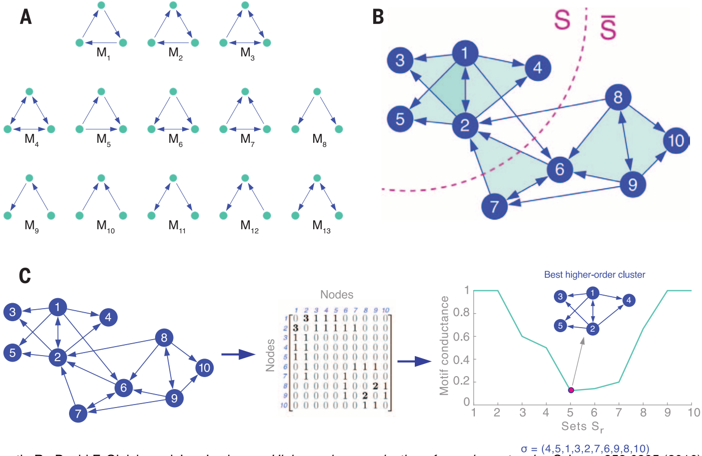
Credit: Wikipedia

Higher-order organization of complex networks

- explores higher-order organization of complex networks at the level of small network subgraphs/motifs
- networks exhibit rich higher-order organizational structures exposed by clustering on higher-order connectivity patterns

Higher-order organization of complex networks

- explores higher-order organization of complex networks at the level of small network subgraphs/motifs
- networks exhibit rich higher-order organizational structures exposed by clustering on higher-order connectivity patterns



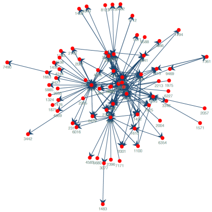
Credit: Benson, Austin R., David F. Gleich, and Jure Leskovec. *Higher-order organization of complex networks.* Science 353.6295 (2016): 163-166.

Core-Periphery Networks

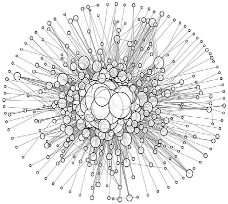
Chaojun Wang, *Core-Periphery Trading Networks*, (2016)

Core-Periphery Networks

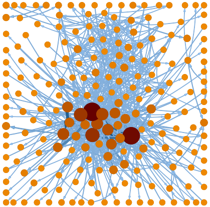
Chaojun Wang, *Core-Periphery Trading Networks*, (2016)



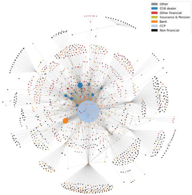
Li and Schüerhoff - muni bonds



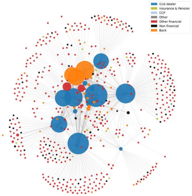
Hollifield, Neklyudov, Spatt - ABS



Bech and Atalay - Fed funds



ESRB - Interest rate swaps



ESRB - Credit default swaps

ESRB - FX forwards

Figure: Core-periphery trading networks in OTC markets.

Core-Periphery structure in (undirected) networks

Structure that consists of densely connected core vertices C and sparsely connected peripheral vertices P (where $V = C \cup P$)

Core-Periphery structure in (undirected) networks

Structure that consists of densely connected core vertices C and sparsely connected peripheral vertices P (where $V = C \cup P$)

- core vertices in C tend to be well-connected both among themselves and to peripheral vertices in P

Core-Periphery structure in (undirected) networks

Structure that consists of densely connected core vertices C and sparsely connected peripheral vertices P (where $V = C \cup P$)

- core vertices in C tend to be well-connected both among themselves and to peripheral vertices in P
- peripheral vertices are sparsely connected to other vertices.

Core-Periphery structure in (undirected) networks

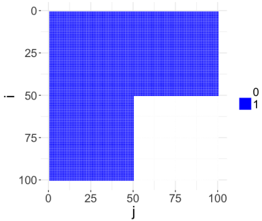
Structure that consists of densely connected core vertices C and sparsely connected peripheral vertices P (where $V = C \cup P$)

- core vertices in C tend to be well-connected both among themselves and to peripheral vertices in P
- peripheral vertices are sparsely connected to other vertices.

A stochastic block model for core-periphery

$$A = \begin{array}{|c|c|}\hline p_{cc} & p_{cp} \\ \hline p_{cp} & p_{pp} \\ \hline \end{array} \tag{41}$$

$$\tilde{A} = \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 0 \\ \hline \end{array} \tag{42}$$



Core-Periphery structure in (undirected) networks

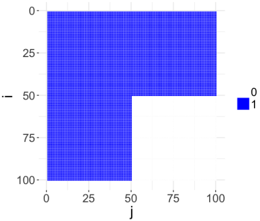
Structure that consists of densely connected core vertices C and sparsely connected peripheral vertices P (where $V = C \cup P$)

- core vertices in C tend to be well-connected both among themselves and to peripheral vertices in P
- peripheral vertices are sparsely connected to other vertices.

A stochastic block model for core-periphery

$$A = \begin{array}{|c|c|} \hline p_{cc} & p_{cp} \\ \hline p_{cp} & p_{pp} \\ \hline \end{array} \tag{41}$$

$$\tilde{A} = \begin{array}{|c|c|} \hline 1 & 1 \\ \hline 1 & 0 \\ \hline \end{array} \tag{42}$$



- $p_{cc} = p_{cp} = 1$ and $p_{pp} = 0$: idealized block model, leading to an adjacency matrix of rank 2

Core-Periphery structure in (undirected) networks

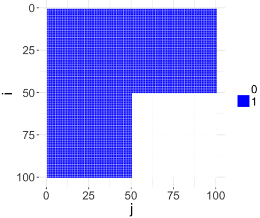
Structure that consists of densely connected core vertices C and sparsely connected peripheral vertices P (where $V = C \cup P$)

- core vertices in C tend to be well-connected both among themselves and to peripheral vertices in P
- peripheral vertices are sparsely connected to other vertices.

A stochastic block model for core-periphery

$$A = \begin{array}{|c|c|} \hline p_{cc} & p_{cp} \\ \hline p_{cp} & p_{pp} \\ \hline \end{array} \qquad (41)$$

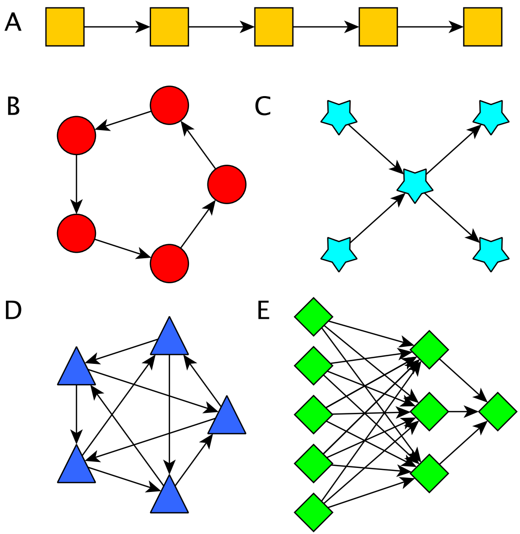
$$\tilde{A} = \begin{array}{|c|c|} \hline 1 & 1 \\ \hline 1 & 0 \\ \hline \end{array} \qquad (42)$$



- $p_{cc} = p_{cp} = 1$ and $p_{pp} = 0$: idealized block model, leading to an adjacency matrix of rank 2
- n_c & n_p the number of core and peripheral vertices; $n_c + n_p = n$
- extension to directed networks: A. Elliott, A. Chiu, M. Bazzi, G. Reinert, M. Cucuringu, *Core-periphery structure in directed networks*, Proc. of the Royal Society A 476, no. 2241

Anomaly detection - identification of heavy structures

The weights of the edges within such structures are considerably larger than the average weight of the ambient graph.



Network Change-point detection

- dynamic networks that are temporal sequences of graph snapshots

Network Change-point detection

- dynamic networks that are temporal sequences of graph snapshots
- goal: detect abrupt changes in their structure

Network Change-point detection

- dynamic networks that are temporal sequences of graph snapshots
- goal: detect abrupt changes in their structure
- cross-correlation networks of daily stock returns (computed over a one-month interval) from the SP 500 index in a period of ~ 20 years (Feb. 2000 - Dec. 2020).
- covariate information available: volatility, volume, etc

Network Change-point detection

- dynamic networks that are temporal sequences of graph snapshots
- goal: detect abrupt changes in their structure
- cross-correlation networks of daily stock returns (computed over a one-month interval) from the SP 500 index in a period of ~ 20 years (Feb. 2000 - Dec. 2020).
- covariate information available: volatility, volume, etc
- move beyond the traditional uni/multi-variate time series for change-point detection (often based on cumsum statistics)

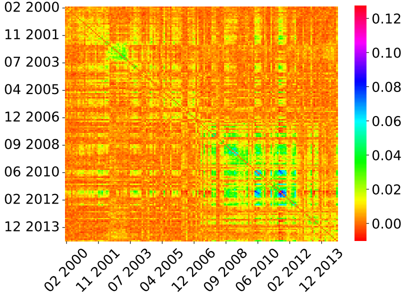


Figure: Matrix of Adjusted Rand Index values between the partitions obtained for each pair of graph snapshots in the correlation network of S&P 500 stock returns. The first two digits denote the month, followed by the year.

Figure: Change-point detection statistics, as obtained by various methods.

Graph Neural Networks

The intuition behind GNN embedding approaches is that at each iteration (search depth, layer) the nodes aggregate information from their local neighbors, and as this process iterates the nodes gather information from further reaches of the graphs.

Graph Neural Networks

The intuition behind GNN embedding approaches is that at each iteration (search depth, layer) the nodes aggregate information from their local neighbors, and as this process iterates the nodes gather information from further reaches of the graphs.

- at each layer k, a representation of each node is available h^k

Graph Neural Networks

The intuition behind GNN embedding approaches is that at each iteration (search depth, layer) the nodes aggregate information from their local neighbors, and as this process iterates the nodes gather information from further reaches of the graphs.

- at each layer k, a representation of each node is available h^k
- each node v AGGREGATEs the representations (embedding) of the nodes in its immediate neighborhood $\{h_u^{k-1}, \forall u \in \mathcal{N}(v)\}$ into a single vector $h_{\mathcal{N}(v)}^{k-1}$ (this depends on the node embeddings from the previous iteration, eg, $k-1$)

Graph Neural Networks

The intuition behind GNN embedding approaches is that at each iteration (search depth, layer) the nodes aggregate information from their local neighbors, and as this process iterates the nodes gather information from further reaches of the graphs.

- at each layer k, a representation of each node is available h^k
- each node v AGGREGATEs the representations (embedding) of the nodes in its immediate neighborhood $\{h_u^{k-1}, \forall u \in \mathcal{N}(v)\}$ into a single vector $h_{\mathcal{N}(v)}^{k-1}$ (this depends on the node embeddings from the previous iteration, eg, $k-1$)
- next, $h_{\mathcal{N}(v)}^{k-1}$ is CONCATenated with h_v^{k-1}, and fed through a fully connected layer with nonlinear activation function σ, whose ouput is $\mapsto h_v^k$

$$h_v^k \leftarrow \sigma\left(W^k \cdot \mathrm{CONCAT}(h_v^{k-1}, h_{\mathcal{N}(v)}^{k-1})\right) \quad (43)$$

Graph Neural Networks

The intuition behind GNN embedding approaches is that at each iteration (search depth, layer) the nodes aggregate information from their local neighbors, and as this process iterates the nodes gather information from further reaches of the graphs.

- at each layer k, a representation of each node is available h^k
- each node v AGGREGATEs the representations (embedding) of the nodes in its immediate neighborhood $\{h_u^{k-1}, \forall u \in \mathcal{N}(v)\}$ into a single vector $h_{\mathcal{N}(v)}^{k-1}$ (this depends on the node embeddings from the previous iteration, eg, $k-1$)
- next, $h_{\mathcal{N}(v)}^{k-1}$ is CONCATenated with h_v^{k-1}, and fed through a fully connected layer with nonlinear activation function σ, whose ouput is $\mapsto h_v^k$

$$h_v^k \leftarrow \sigma\left(W^k \cdot \mathrm{CONCAT}(h_v^{k-1}, h_{\mathcal{N}(v)}^{k-1})\right) \tag{43}$$

- in order to learn useful, predictive representations in a fully unsupervised setting, one can apply a graph-based loss function to the final output embeddings z_u, and tune the weight matrices W^k and parameters of the AGGREGATE function via stochastic gradient descent.

Graph Neural Networks

The intuition behind GNN embedding approaches is that at each iteration (search depth, layer) the nodes aggregate information from their local neighbors, and as this process iterates the nodes gather information from further reaches of the graphs.

- at each layer k, a representation of each node is available h^k
- each node v AGGREGATEs the representations (embedding) of the nodes in its immediate neighborhood $\{h_u^{k-1}, \forall u \in \mathcal{N}(v)\}$ into a single vector $h_{\mathcal{N}(v)}^{k-1}$ (this depends on the node embeddings from the previous iteration, eg, $k-1$)
- next, $h_{\mathcal{N}(v)}^{k-1}$ is CONCATenated with h_v^{k-1}, and fed through a fully connected layer with nonlinear activation function σ, whose ouput is $\mapsto h_v^k$

$$h_v^k \leftarrow \sigma\left(W^k \cdot \text{CONCAT}(h_v^{k-1}, h_{\mathcal{N}(v)}^{k-1})\right) \tag{43}$$

- in order to learn useful, predictive representations in a fully unsupervised setting, one can apply a graph-based loss function to the final output embeddings z_u, and tune the weight matrices W^k and parameters of the AGGREGATE function via stochastic gradient descent.

Remarks

- a graph-based loss function promotes nearby nodes to have similar representations, while enforcing that the representations of far-away nodes are highly distinct

Graph Neural Networks

The intuition behind GNN embedding approaches is that at each iteration (search depth, layer) the nodes aggregate information from their local neighbors, and as this process iterates the nodes gather information from further reaches of the graphs.

- at each layer k, a representation of each node is available h^k
- each node v AGGREGATEs the representations (embedding) of the nodes in its immediate neighborhood $\{h_u^{k-1}, \forall u \in \mathcal{N}(v)\}$ into a single vector $h_{\mathcal{N}(v)}^{k-1}$ (this depends on the node embeddings from the previous iteration, eg, $k-1$)
- next, $h_{\mathcal{N}(v)}^{k-1}$ is CONCATenated with h_v^{k-1}, and fed through a fully connected layer with nonlinear activation function σ, whose ouput is $\mapsto h_v^k$

$$h_v^k \leftarrow \sigma\left(W^k \cdot \mathrm{CONCAT}(h_v^{k-1}, h_{\mathcal{N}(v)}^{k-1})\right) \tag{43}$$

- in order to learn useful, predictive representations in a fully unsupervised setting, one can apply a graph-based loss function to the final output embeddings z_u, and tune the weight matrices W^k and parameters of the AGGREGATE function via stochastic gradient descent.

Remarks

- a graph-based loss function promotes nearby nodes to have similar representations, while enforcing that the representations of far-away nodes are highly distinct
- when the representations are used for a specific downstream task, the unsupervised loss above is replaced/augmented by a task-specific objective

Outlook - active research areas on graph-based problems

- How to formalize this notion of (directed) communities beyond stochastic block models? (i.e., similar to NCUT in undirected graphs)

Outlook - active research areas on graph-based problems

- How to formalize this notion of (directed) communities beyond stochastic block models? (i.e., similar to NCUT in undirected graphs)
- How to combine patterns in edge directions with different edge densities?

Outlook - active research areas on graph-based problems

- How to formalize this notion of (directed) communities beyond stochastic block models? (i.e., similar to NCUT in undirected graphs)
- How to combine patterns in edge directions with different edge densities?
- Extension to the time dependent setting - what are good probabilistic models to track temporal changes in a network?

Outlook - active research areas on graph-based problems

- How to formalize this notion of (directed) communities beyond stochastic block models? (i.e., similar to NCUT in undirected graphs)
- How to combine patterns in edge directions with different edge densities?
- Extension to the time dependent setting - what are good probabilistic models to track temporal changes in a network?
- Change-point detection in network time series data

Outlook - active research areas on graph-based problems

- How to formalize this notion of (directed) communities beyond stochastic block models? (i.e., similar to NCut in undirected graphs)
- How to combine patterns in edge directions with different edge densities?
- Extension to the time dependent setting - what are good probabilistic models to track temporal changes in a network?
- Change-point detection in network time series data
- Extend theory to general $k > 2$ for more general SSBM (unequal clusters, different p, η)

Outlook - active research areas on graph-based problems

- How to formalize this notion of (directed) communities beyond stochastic block models? (i.e., similar to NCUT in undirected graphs)
- How to combine patterns in edge directions with different edge densities?
- Extension to the time dependent setting - what are good probabilistic models to track temporal changes in a network?
- Change-point detection in network time series data
- Extend theory to general $k > 2$ for more general SSBM (unequal clusters, different p, η)
- Extend methods to the sparse regime $p = \Theta(1/n)$ (current theory fails). What are good regularization approaches for signed graphs? Directed?

Outlook - active research areas on graph-based problems

- How to formalize this notion of (directed) communities beyond stochastic block models? (i.e., similar to NCUT in undirected graphs)
- How to combine patterns in edge directions with different edge densities?
- Extension to the time dependent setting - what are good probabilistic models to track temporal changes in a network?
- Change-point detection in network time series data
- Extend theory to general $k > 2$ for more general SSBM (unequal clusters, different p, η)
- Extend methods to the sparse regime $p = \Theta(1/n)$ (current theory fails). What are good regularization approaches for signed graphs? Directed?
- Graph neural network (GNN)-based methods for signed/directed graphs?

Outlook - active research areas on graph-based problems

- How to formalize this notion of (directed) communities beyond stochastic block models? (i.e., similar to NCUT in undirected graphs)
- How to combine patterns in edge directions with different edge densities?
- Extension to the time dependent setting - what are good probabilistic models to track temporal changes in a network?
- Change-point detection in network time series data
- Extend theory to general $k > 2$ for more general SSBM (unequal clusters, different p, η)
- Extend methods to the sparse regime $p = \Theta(1/n)$ (current theory fails). What are good regularization approaches for signed graphs? Directed?
- Graph neural network (GNN)-based methods for signed/directed graphs?
- Extension to hypergraphs and higher-order structures

Outlook - active research areas on graph-based problems

- How to formalize this notion of (directed) communities beyond stochastic block models? (i.e., similar to NCUT in undirected graphs)
- How to combine patterns in edge directions with different edge densities?
- Extension to the time dependent setting - what are good probabilistic models to track temporal changes in a network?
- Change-point detection in network time series data
- Extend theory to general $k > 2$ for more general SSBM (unequal clusters, different p, η)
- Extend methods to the sparse regime $p = \Theta(1/n)$ (current theory fails). What are good regularization approaches for signed graphs? Directed?
- Graph neural network (GNN)-based methods for signed/directed graphs?
- Extension to hypergraphs and higher-order structures
- Explore applications to finance and cyber security, in terms of prediction (next lecture) and anomaly detection.

Relevant literature for applications of networks in finance

1. *“Sentiment Correlation in Financial News Networks and Associated Market Movements"*, Wan, Yang, Marinov, Calliess, Zohren, Dong, Scientific Reports 11, 3062 (2021)
2. *“Temporal Graph Networks for Deep Learning on Dynamic Graphs"*, Rossi, Chamberlain, Frasca, Eynard, Monti, Bronstein, `https://arxiv.org/abs/2006.10637`
3. *“Topological structures in the equities market network"*, Gregory Leibon, Scott Pauls, Daniel Rockmore, and Robert Savell, PNAS 2008, 105 (52) 20589-20594
4. *“Modeling the Stock Relation with Graph Networks for Overnight Stock Movement Prediction"*, IJCAI-20, `https://www.ijcai.org/Proceedings/2020/0626.pdf`
5. *“Modeling the Momentum Spillover Effect for Stock Prediction via Attribute-Driven Graph Attention Networks"*, AAAI Conference on Artificial Intelligence, 35(1), 55-62
6. *“Knowledge Graph-based Event Embedding Framework for Financial Quantitative Investments"*, Cheng, Yang, Wang, Zhang, Zhang, SIGIR 2020
7. *“Analysis of Equity Markets: A Graph Theory Approach analysis of equity markets a graph theory approach"*, `https://evoq-eval.siam.org/Portals/0/Publications/SIURO/Volume%2010/Analysis_Equity_Markets_A_Graph_Theory_Approach.pdf?ver=2018-02-28-145946-083`
8. *“Stock Network Stability After Crashes Based on Entropy Method"*, Front. Phys., 12 June 2020 `https://doi.org/10.3389/fphy.2020.00163`
9. *“Stock market network's topological stability: Evidence from planar maximally filtered graph and minimal spanning tree"*, International Journal of Modern Physics B (2015)
10. *“Correlation based networks of equity returns sampled at different time horizons"* `https://arxiv.org/pdf/physics/0605251.pdf`
11. *“Stability Analysis of Company Co-Mention Network and Market Graph Over Time Using Graph Similarity Measures"*, J. Open Innov. Technol. Mark. Complex. 2019

Lead-lag detection in multivariate time series

- (Global) Ranking-based lead-lag
- Local/Partial Ranking-based lead-lag
- Cluster-based lead-lag
- Ranking from pairwise comparisons

1

Leaders and laggers in multivariate time series

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

1

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 economics, earth science, biology

1

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 economics, earth science, biology
- finance: lead-lag dependence between the spot and futures markets for a Chinese stock market index (Wang et al. 2017)

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 economics, earth science, biology
- finance: lead-lag dependence between the spot and futures markets for a Chinese stock market index (Wang et al. 2017)
- 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 economics, earth science, biology
- finance: lead-lag dependence between the spot and futures markets for a Chinese stock market index (Wang et al. 2017)
- 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 economics, earth science, biology
- finance: lead-lag dependence between the spot and futures markets for a Chinese stock market index (Wang et al. 2017)
- 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)

1

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 economics, earth science, biology
- finance: lead-lag dependence between the spot and futures markets for a Chinese stock market index (Wang et al. 2017)
- 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

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 economics, earth science, biology
- finance: lead-lag dependence between the spot and futures markets for a Chinese stock market index (Wang et al. 2017)
- 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

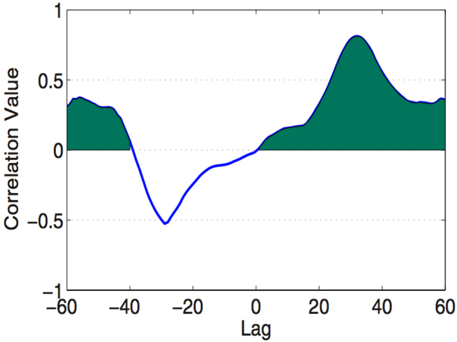
Leaders and laggers in multivariate time series

- 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 economics, earth science, biology
- finance: lead-lag dependence between the spot and futures markets for a Chinese stock market index (Wang et al. 2017)
- 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

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 economics, earth science, biology
- finance: lead-lag dependence between the spot and futures markets for a Chinese stock market index (Wang et al. 2017)
- 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

Cross-correlations and the lead-lag matrix

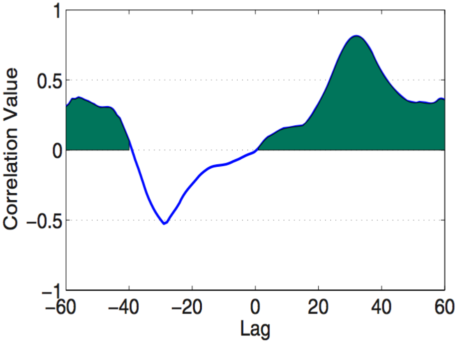


Wu et al, 2010

Options for building the pairwise comparison matrix:

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

Cross-correlations and the lead-lag matrix

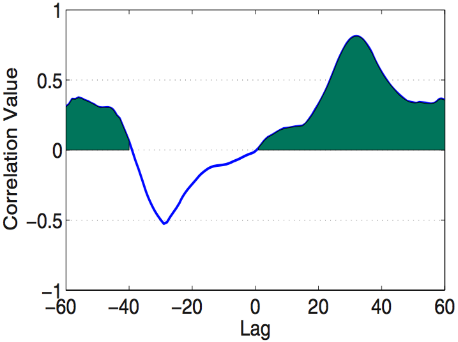


Wu et al, 2010

Options for building the pairwise comparison matrix:

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

Cross-correlations and the lead-lag matrix



Wu et al, 2010

Options for building the 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

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

Lead-lag detection in multivariate time series

- (Global) Ranking-based lead-lag
- Local/Partial Ranking-based lead-lag
- Cluster-based lead-lag
- Ranking from pairwise comparisons

(1) Global ranking of the time series

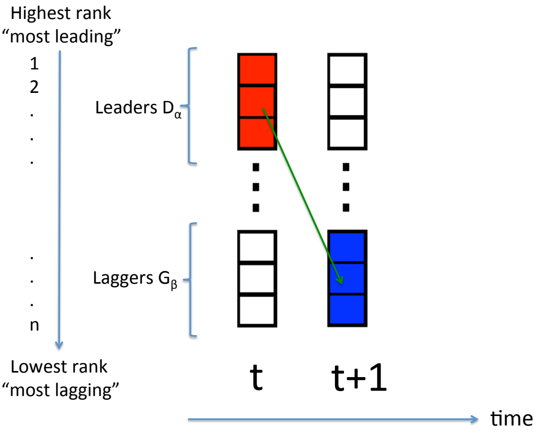
- find a global ranking of the time series

(1) Global ranking of the time series

- find a global ranking of the time series
- construct a leading and a lagging cluster

(1) Global ranking of the time series

- find a global ranking of the time series
- construct a leading and a lagging cluster
- build a forecast for the lagging cluster catching up to the leading cluster



Ranking time series (sector ETFs)

Study with a small universe of instruments (sector ETFs in the US equity market).

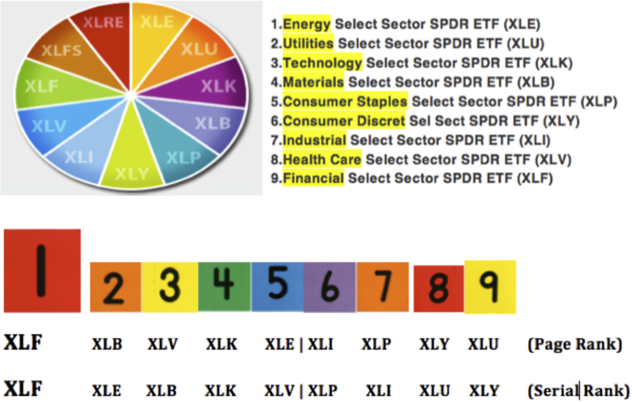


Figure: US Sector ETFs

Student class project (UCLA).

Data set 1: S&P 500 constituents (470)

- 2003-2014, 3000+ trading days
- daily log returns
- on any given day, use the past $m = 60$ days of historical data

Data set 1: S&P 500 constituents (470)

- 2003-2014, 3000+ trading days
- daily log returns
- on any given day, use the past $m = 60$ days of historical data

- forecast the future 1-day return

Data set 1: S&P 500 constituents (470)

- 2003-2014, 3000+ trading days
- daily log returns
- on any given day, use the past $m = 60$ days of historical data
- forecast the future 1-day return
- universe given by S&P 500 constituents (and 10 sector ETFs)

Data set 1: S&P 500 constituents (470)

- 2003-2014, 3000+ trading days
- daily log returns
- on any given day, use the past $m = 60$ days of historical data
- forecast the future 1-day return
- universe given by S&P 500 constituents (and 10 sector ETFs)
- 1-to-1 hedge with {SPY, the basket of leaders}

Data set 1: S&P 500 constituents (470)

- 2003-2014, 3000+ trading days
- daily log returns
- on any given day, use the past $m = 60$ days of historical data

- forecast the future 1-day return

- universe given by S&P 500 constituents (and 10 sector ETFs)

- 1-to-1 hedge with {SPY, the basket of leaders}

Note: simple momentum-based approach ($D = G = [n]$) yields

- Sharpe = 0.5
- P&L = 1 bpts/day

Data set 1: S&P 500 constituents (470)

- 2003-2014, 3000+ trading days
- daily log returns
- on any given day, use the past $m = 60$ days of historical data
- forecast the future 1-day return
- universe given by S&P 500 constituents (and 10 sector ETFs)
- 1-to-1 hedge with {SPY, the basket of leaders}

Note: simple momentum-based approach ($D = G = [n]$) yields

- Sharpe = 0.5
- P&L = 1 bpts/day

Keep $\beta = 1 - \alpha$ fixed (could replace by a ”Cheeger sweep")

SP500 universe: P&L across time (2003-2014)

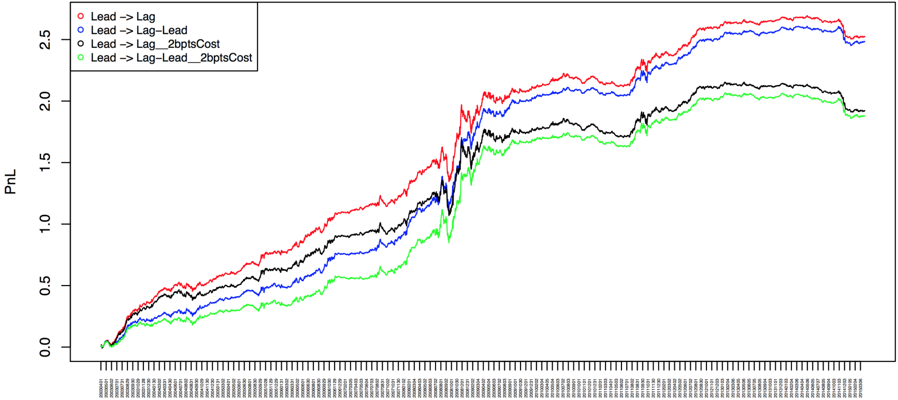


Figure: β = 0.10%, α = 0.90%

Performance statistics: Sharpe Ratio, P&L

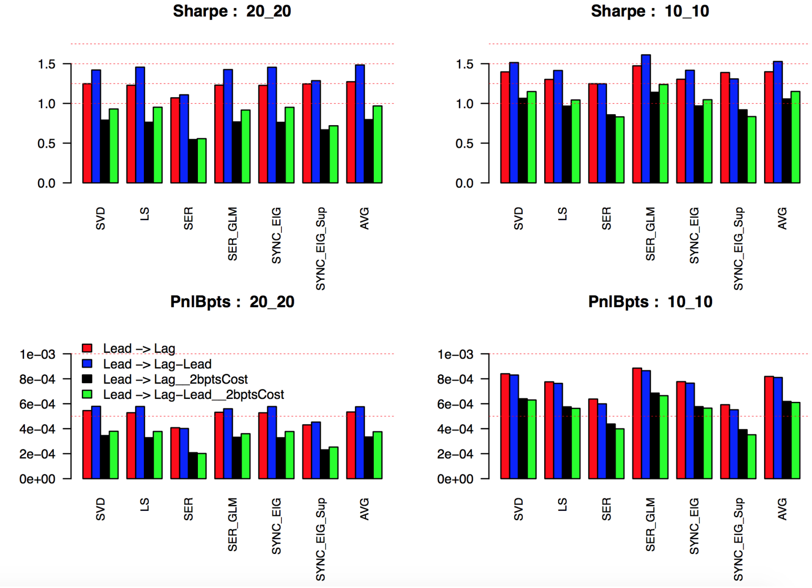
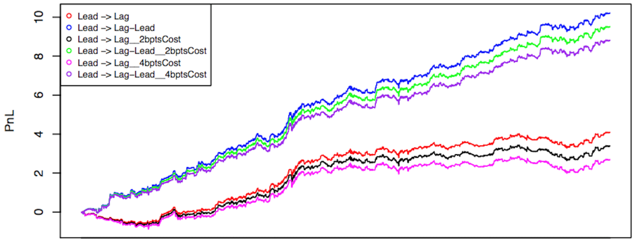


Figure: Left: (β = 20%, α = 80%). Right: (β = 10%, α = 90%)

Data set 2: Macro ETFs (n = 10)

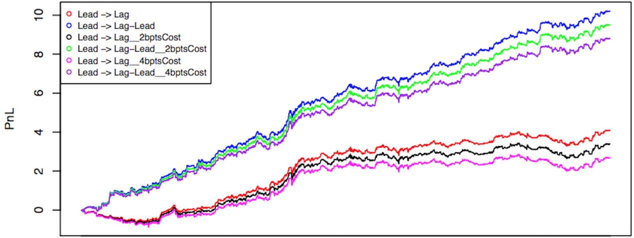
- 2003-2017, 3500+ trading days
- universe of size 10 instruments: SPY, FTSE, N225, IWM, EEM, XLF, VIX, TLT, USO, GLD
- 1-to-1 hedge with {SPY, the basket of leaders}

P&L for top performing ranking methods (basket of size 5)

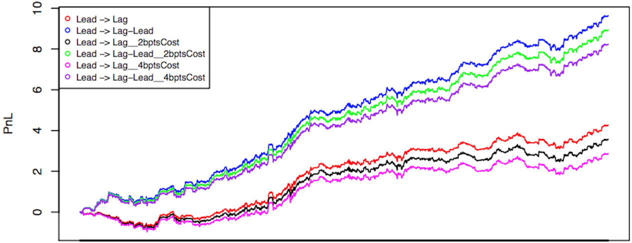


Serial-Rank

P&L for top performing ranking methods (basket of size 5)



Serial-Rank



Sync-Rank-Sup

Figure: $\beta = 0.5$, $\alpha = 0.5$

Modularity clustering of the residual matrix; S&P 500

The residual matrix $R = |C - \hat{C}|$.

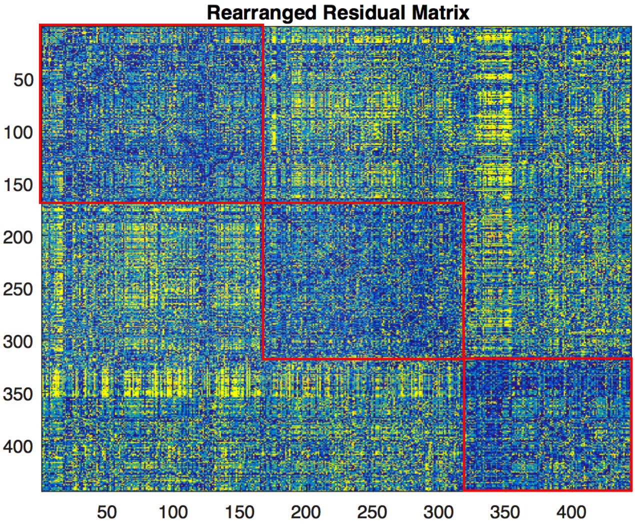


Figure: Blue denotes low absolute values. Yellow denotes high magnitude values.

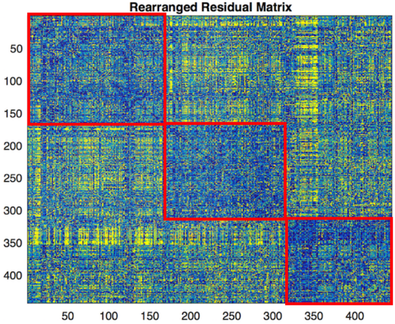
(2) Cluster and extract partial rankings

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

(2) Cluster and extract 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



η_1	η_2	η_2
η_2	η_1	η_2
η_2	η_2	η_1

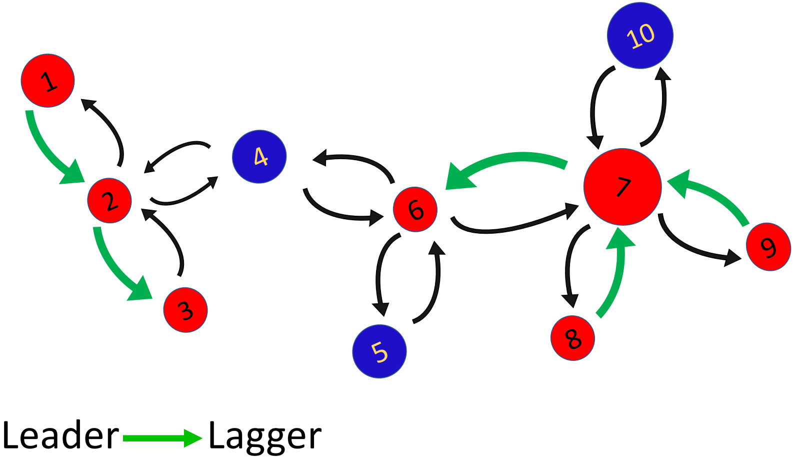
Noise levels $\eta_1 < \eta_2$

- Stochastic block model for ranking; heterogeneous noise (easier to solve (*rank*) within each cluster)

Lead-lag detection in multivariate time series

- (Global) Ranking-based lead-lag
- Local/Partial Ranking-based lead-lag
- Cluster-based lead-lag
- Ranking from pairwise comparisons

(3) Uncover underlying directed meta-graph structure



- *"Detection and clustering of lead-lag networks for multivariate time series with an application to financial markets"*, Bennett, Cucuringu and Reinert
- 7th Workshop on Mining and Learning from Time Series (MiLeTS), KDD 2021, poster & spotlight presentation

Lead-lag detection and network clustering for multivariate time series with an application to the US equity market, Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert (KDD 2021 + under review at journal)

Main contributions:

- introduction of a principled method to address the problem of unsupervised clustering of leading and lagging variables in multivariate time series

Lead-lag detection and network clustering for multivariate time series with an application to the US equity market, Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert (KDD 2021 + under review at journal)

Main contributions:

- introduction of a principled method to address the problem of unsupervised clustering of leading and lagging variables in multivariate time series
    - validate different components on synthetic and real data sets

Lead-lag detection and network clustering for multivariate time series with an application to the US equity market, Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert (KDD 2021 + under review at journal)

Main contributions:

- introduction of a principled method to address the problem of unsupervised clustering of leading and lagging variables in multivariate time series
    - validate different components on synthetic and real data sets
- evaluation of novel pairwise lead-lag metrics using a new benchmark data generating process for multivariate time series systems with clustered lead-lag structure.

Lead-lag detection and network clustering for multivariate time series with an application to the US equity market, Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert (KDD 2021 + under review at journal)

Main contributions:

- introduction of a principled method to address the problem of unsupervised clustering of leading and lagging variables in multivariate time series
  - validate different components on synthetic and real data sets
- evaluation of novel pairwise lead-lag metrics using a new benchmark data generating process for multivariate time series systems with clustered lead-lag structure.
- application of our method to US equity data provides insights into the structure of the US equity market.

Lead-lag detection and network clustering for multivariate time series with an application to the US equity market, Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert (KDD 2021 + under review at journal)

Main contributions:

- introduction of a principled method to address the problem of unsupervised clustering of leading and lagging variables in multivariate time series
  - validate different components on synthetic and real data sets
- evaluation of novel pairwise lead-lag metrics using a new benchmark data generating process for multivariate time series systems with clustered lead-lag structure.
- application of our method to US equity data provides insights into the structure of the US equity market.

To the best of our knowledge, the first data-driven clustering of lead-lag networks in a financial market context.

Data description

- universe of 5325 NYSE equities spanning from 04-01-2000 to 31-12-2019

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

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

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

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

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

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 & leadingness metric

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

Cluster meta-flow matrix & leadingness metric

- total *flow* between 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 & leadingness metric

- total *flow* between 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 & leadingness metric

- total *flow* between 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 $i \in \{1, \ldots, k\}$

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

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 & leadingness metric

- total *flow* between 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 $i \in \{1, \ldots, k\}$

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

  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 & leadingness metric

- total *flow* between 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 $i \in \{1, \ldots, k\}$

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

  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) (labeled $k - 1$)

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

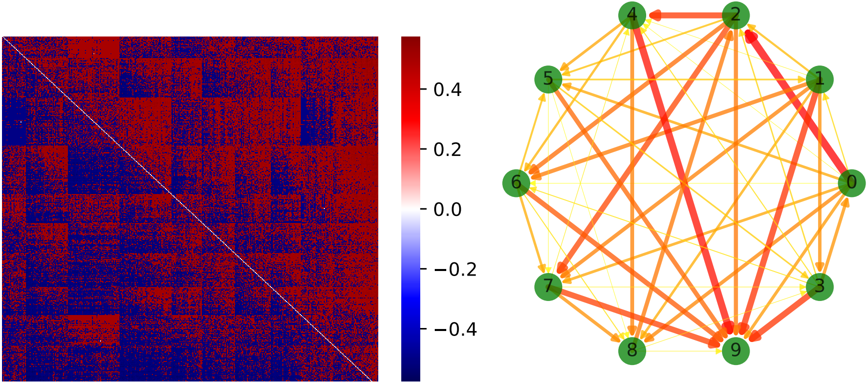


Figure: Left: Heatmap of the double-sorted lead-lag $n \times n$ matrix $A - A^T$. The rows and columns of the matrix index the $n = 434$ equities, and are categorised by cluster membership (labelled by the leadingness metric. Within each cluster, we sort the equities by their respective row-sum in $A - A^T$, 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.

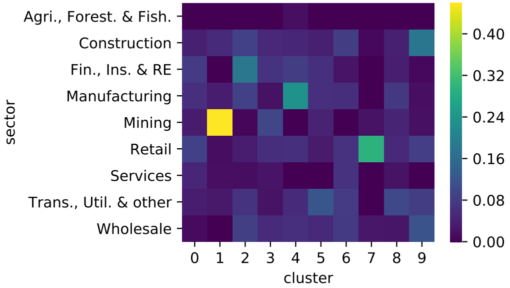


Figure: The Jaccard similarity coefficient between the Hermitian RW clusters and industry clusters. Cluster 0 is most leading; cluster 9 is most lagging.

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.

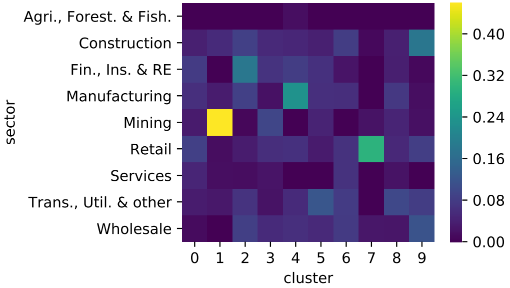


Figure: The Jaccard similarity coefficient between the Hermitian RW clusters and industry clusters. Cluster 0 is most leading; cluster 9 is most lagging.

- test for a statistically significant time dependence in daily returns using a permutation test on the spectrum of the Hermitian adj. mtx. $\tilde{A} = i(A - A^T)$

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.

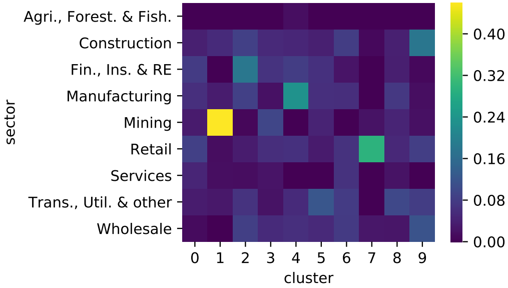


Figure: The Jaccard similarity coefficient between the Hermitian RW clusters and industry clusters. Cluster 0 is most leading; cluster 9 is most lagging.

- 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 $\tilde{A}$

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.

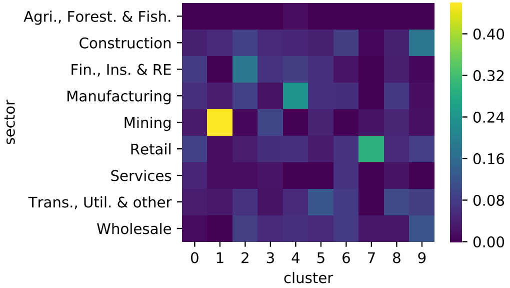


Figure: The Jaccard similarity coefficient between the Hermitian RW clusters and industry clusters. Cluster 0 is most leading; cluster 9 is most lagging.

- 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

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

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

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

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

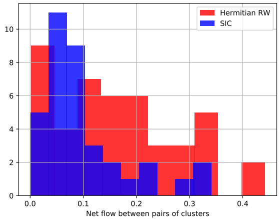
1. Sector membership induces clustered lead-lag effects
   - Biely and Thurner [2008] fin find associations between sector membership and lead-lag structure on the high-frequency scale of returns
2. Equities with higher trading volume are hypothesized to lead lower volume equities
   - disparities in 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.

Data-driven clustering with known lead-lag mechanisms

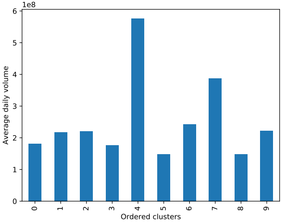
Recovered clusterings **cannot** be explained by the three previously hypothesized mechanisms in the empirical finance lead-lag literature

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.

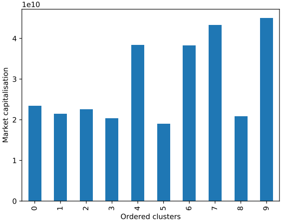
Data-driven clustering with known lead-lag mechanisms



(a) Histogram 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.

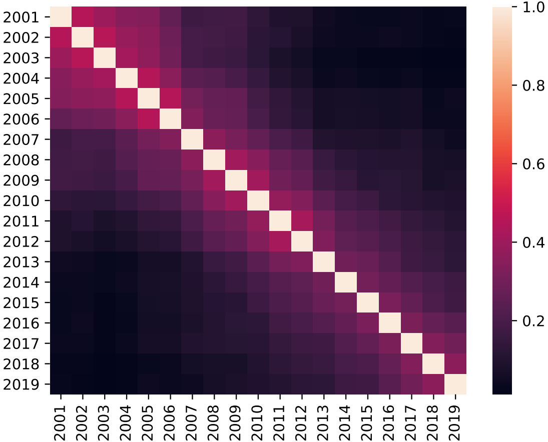


Figure: Adjusted Rand index between clusters computed on yearly snapshots of data

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

Cumulative profit (P&L) of the signal

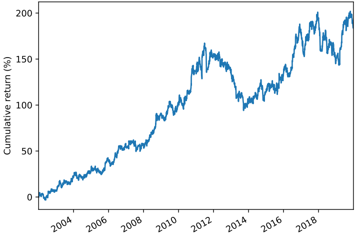


Figure: Cumulative return for the financial forecasting signal.

- 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.

Cumulative profit (P&L) of the signal

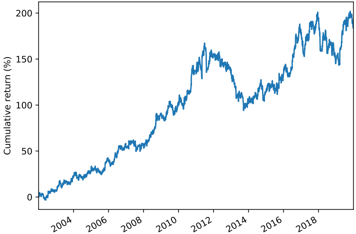


Figure: Cumulative return for the financial forecasting signal.

- 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

Cumulative profit (P&L) of the signal

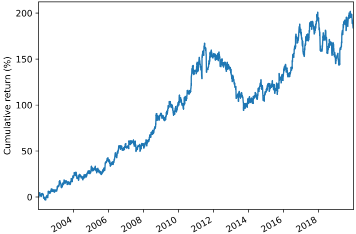


Figure: Cumulative return for the financial forecasting signal.

- 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

Cumulative profit (P&L) of the signal

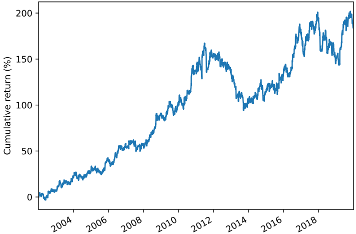


Figure: Cumulative return for the financial forecasting signal.

- 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 daily return of the trading signal is 2.4 basis points

Cumulative profit (P&L) of the signal

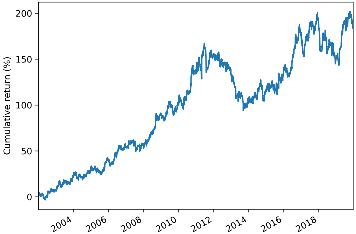


Figure: Cumulative return for the financial forecasting signal.

- 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 daily return of the trading signal is 2.4 basis points
- decay in the performance of the signal after 2012;

Cumulative profit (P&L) of the signal

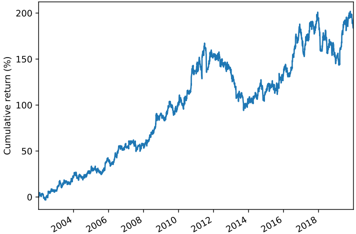


Figure: Cumulative return for the financial forecasting signal.

- 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 daily return of the trading signal is 2.4 basis points
- decay in the performance of the signal after 2012;
- can be compared with the reduction in clustering persistence post 2012

Cumulative profit (P&L) of the signal

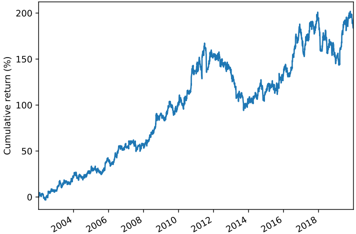
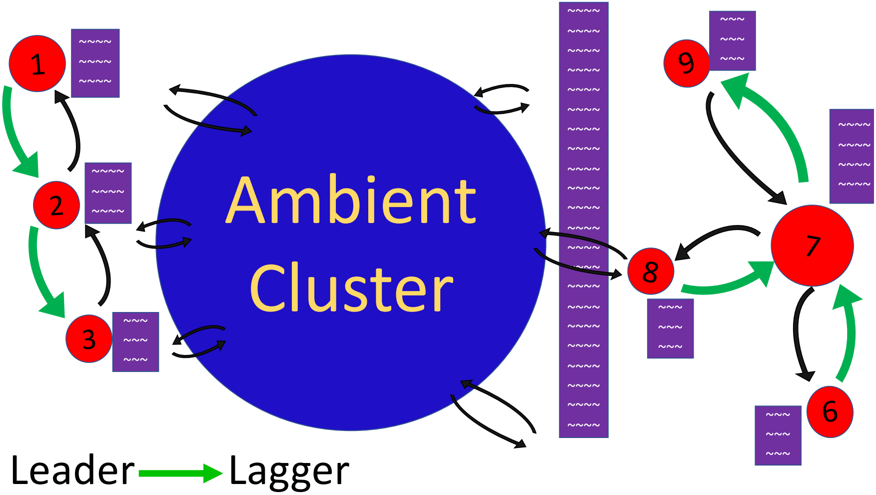


Figure: Cumulative return for the financial forecasting signal.

- 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 daily return of the trading signal is 2.4 basis points
- 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

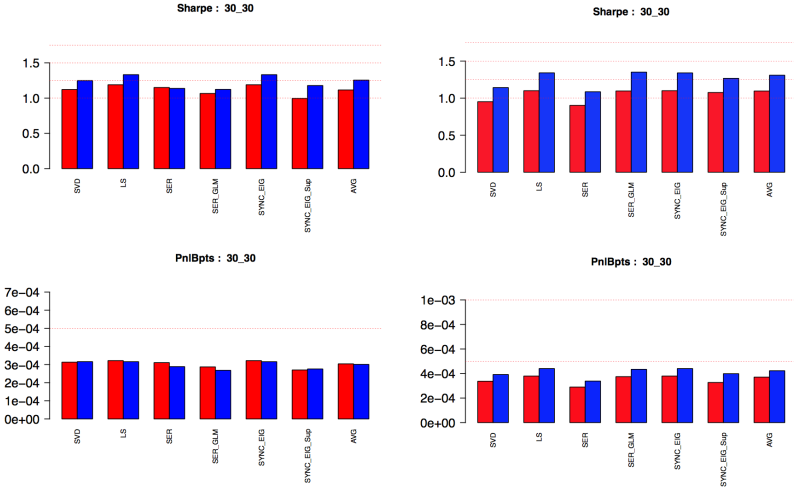
Planted structure in a much larger ambient graph + Handling node level covariates



Motivates the use Graph Neural Networks.

Signatures make a difference

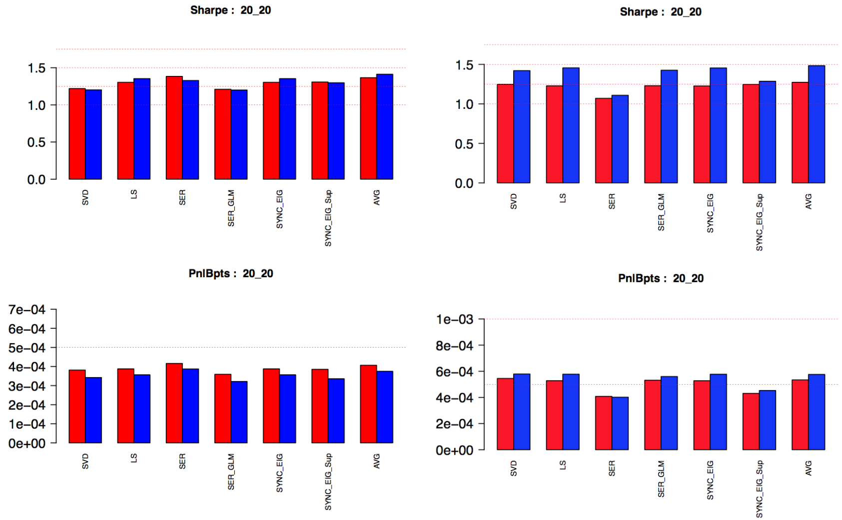
- Back in the global ranking setup, for the SP 500 data set.



(a) Avg Max Corr

(b) Signatures

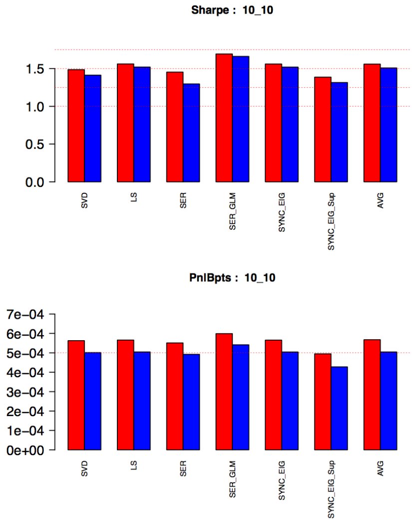
Figure: Bottom β = 30% laggers (α = 70%) (SP500 2003-2015). Colors are different markouts: trading the laggers only (red), trading the spread laggers - leaders (blue).



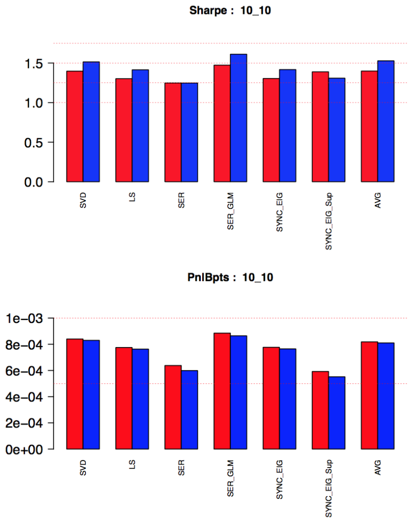
(a) Avg Max Corr

(b) Signatures

Figure: Bottom β = 20% laggers (α = 80%) (SP500 2003-2015)



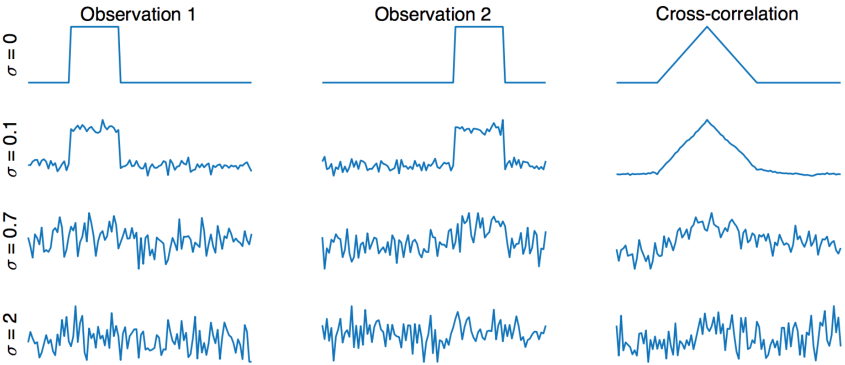
(a) Avg Max Corr



(b) Signatures

Figure: Bottom β = 10% laggers (α = 90%) (SP500 2003-2015)

Ongoing work: connections with multi-reference alignment



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

Models

Multi-reference alignment:

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

Models

Multi-reference alignment:

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

Models

Multi-reference alignment:

- 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

Models

Multi-reference alignment:

- 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

Models

Multi-reference alignment:

- 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

Models

Multi-reference alignment:

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

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
- $r = Lf + \epsilon$

Models

Multi-reference alignment:

- 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)

Models

Multi-reference alignment:

- 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

$$
L = \begin{bmatrix}
1 & 0 & 0 \\
1 & 0 & 0 \\
\vdots & \vdots & \vdots \\
1 & 0 & 0 \\
1 & 0 & 0 \\
\hline
0 & 1 & 0 \\
0 & 1 & 0 \\
\vdots & \vdots & \vdots \\
0 & 1 & 0 \\
0 & 1 & 0 \\
\hline
0 & 0 & 1 \\
0 & 0 & 1 \\
\vdots & \vdots & \vdots \\
0 & 0 & 1 \\
0 & 0 & 1
\end{bmatrix}
\qquad
G = \begin{bmatrix}
3 & 0 & 0 \\
1 & 0 & 0 \\
\vdots & \vdots & \vdots \\
m & 0 & 0 \\
2 & 0 & 0 \\
\hline
0 & 5 & 0 \\
0 & m & 0 \\
\vdots & \vdots & \vdots \\
0 & 1 & 0 \\
0 & 3 & 0 \\
\hline
0 & 0 & 3 \\
0 & 0 & 2 \\
\vdots & \vdots & \vdots \\
0 & 0 & 4 \\
0 & 0 & m
\end{bmatrix}
$$

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

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad G = \begin{bmatrix} 3 & 0 & 0 \\ 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ m & 0 & 0 \\ 2 & 0 & 0 \\ \hline 0 & 5 & 0 \\ 0 & m & 0 \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \\ 0 & 3 & 0 \\ \hline 0 & 0 & 3 \\ 0 & 0 & 2 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 4 \\ 0 & 0 & m \end{bmatrix}$$

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 (signed) clustering to propose an algorithm that recovers the matrix G, and leverage this for prediction tasks.

Lead-lag detection in multivariate time series

- (Global) Ranking-based lead-lag
- Local/Partial Ranking-based lead-lag
- Cluster-based lead-lag
- Ranking from pairwise comparisons

Ranking from pairwise information

n players: incomplete inconsistent pairwise comparisons

$$\text{(ordinal)} \qquad \text{Player}_i \succ \text{Player}_j$$

$$\text{(cardinal)} \qquad \text{Player}_i \;\; 3 \; : \; 1 \;\; \text{Player}_j$$

Goal: infer a global or partial ranking $\pi(i)$ of the n players

$$\text{Player}_{\pi(1)} \succ \text{Player}_{\pi(2)} \succ \ldots \succ \text{Player}_{\pi(n)}$$

that best agrees with the data (minimize the number of upsets)

Ranking from pairwise information

n players: incomplete inconsistent pairwise comparisons

$$\text{(ordinal)} \qquad \text{Player}_i \succ \text{Player}_j$$

$$\text{(cardinal)} \qquad \text{Player}_i \;\; 3 \; : \; 1 \;\; \text{Player}_j$$

Goal: infer a global or partial ranking $\pi(i)$ of the n players

$$\text{Player}_{\pi(1)} \succ \text{Player}_{\pi(2)} \succ \ldots \succ \text{Player}_{\pi(n)}$$

that best agrees with the data (minimize the number of upsets)

- $r_1, r_2, \ldots, r_n$ ground truth ranking (or strengths $\in \mathbb{R}$)
- available pairwise comparisons are a proxy for $r_i - r_j$

Ranking from pairwise information

n players: incomplete inconsistent pairwise comparisons

$$\text{(ordinal)} \qquad \text{Player}_i \succ \text{Player}_j$$

$$\text{(cardinal)} \qquad \text{Player}_i \ 3 : 1 \ \text{Player}_j$$

Goal: infer a global or partial ranking $\pi(i)$ of the n players

$$\text{Player}_{\pi(1)} \succ \text{Player}_{\pi(2)} \succ \ldots \succ \text{Player}_{\pi(n)}$$

that best agrees with the data (minimize the number of upsets)

- $r_1, r_2, \ldots, r_n$ ground truth ranking (or strengths $\in \mathbb{R}$)
- available pairwise comparisons are a proxy for $r_i - r_j$
- in most practical applications, the available information
  - is usually incomplete, especially when n is large
  - is very noisy (inconsistent with an underlying total ordering)
- $G = K_n$: cycles, A beats B, B beats C, and C beats A

Ranking from pairwise information

n players: incomplete inconsistent pairwise comparisons

$$\text{(ordinal)} \qquad \text{Player}_i \succ \text{Player}_j$$

$$\text{(cardinal)} \qquad \text{Player}_i \;\; 3 \; : \; 1 \;\; \text{Player}_j$$

Goal: infer a global or partial ranking $\pi(i)$ of the n players

$$\text{Player}_{\pi(1)} \succ \text{Player}_{\pi(2)} \succ \ldots \succ \text{Player}_{\pi(n)}$$

that best agrees with the data (minimize the number of upsets)

- $r_1, r_2, \ldots, r_n$ ground truth ranking (or strengths $\in \mathbb{R}$)
- available pairwise comparisons are a proxy for $r_i - r_j$
- in most practical applications, the available information
  - is usually incomplete, especially when n is large
  - is very noisy (inconsistent with an underlying total ordering)
- $G = K_n$: cycles, A beats B, B beats C, and C beats A

ERO(n, p, η): **Erdős-Rényi Outliers** noise model

$$C_{ij} = \begin{cases} r_i - r_j & \text{correct edge} & \text{w.p. } (1-\eta)p \\ \sim \text{Unif}[-(n-1), n-1] & \text{incorrect edge} & \text{w.p. } \eta p \\ 0 & \text{missing edge,} & \text{w.p. } 1-p \end{cases} \qquad (2)$$

Singular Value Decomposition (SVD) ranking

- for cardinal measurements $C_{ij} = r_i - r_j$, the noiseless matrix of rank offsets $C = (C_{ij})_{1 \le i,j \le n}$ is a skew-symmetric of even rank 2

Singular Value Decomposition (SVD) ranking

- for cardinal measurements $C_{ij} = r_i - r_j$, the noiseless matrix of rank offsets $C = (C_{ij})_{1 \leq i,j \leq n}$ is a skew-symmetric of even rank 2

$$C = r\boldsymbol{e}^T - \boldsymbol{e}r^T \tag{3}$$

where $\boldsymbol{e}$ denotes the all-ones column vector (check this!)

Singular Value Decomposition (SVD) ranking

- for cardinal measurements $C_{ij} = r_i - r_j$, the noiseless matrix of rank offsets $C = (C_{ij})_{1 \le i,j \le n}$ is a skew-symmetric of even rank 2
$$C = r\boldsymbol{e}^T - \boldsymbol{e}r^T \tag{3}$$
where $\boldsymbol{e}$ denotes the all-ones column vector (check this!)
- ERO noise model
$$\mathbb{E}C_{ij} = (r_i - r_j)(1 - \eta)p, \tag{4}$$

Singular Value Decomposition (SVD) ranking

- for cardinal measurements $C_{ij} = r_i - r_j$, the noiseless matrix of rank offsets $C = (C_{ij})_{1 \leq i,j \leq n}$ is a skew-symmetric of even rank 2
$$C = r\boldsymbol{e}^T - \boldsymbol{e}r^T \tag{3}$$
where $\boldsymbol{e}$ denotes the all-ones column vector (check this!)
- ERO noise model
$$\mathbb{E}C_{ij} = (r_i - r_j)(1 - \eta)p, \tag{4}$$
- $\mathbb{E}C$ is a rank-2 skew-symmetric matrix
$$\mathbb{E}C = (1 - \eta)p(r\boldsymbol{e}^T - \boldsymbol{e}r^T) \tag{5}$$

Singular Value Decomposition (SVD) ranking

- for cardinal measurements $C_{ij} = r_i - r_j$, the noiseless matrix of rank offsets $C = (C_{ij})_{1 \le i,j \le n}$ is a skew-symmetric of even rank 2
$$C = r\boldsymbol{e}^T - \boldsymbol{e}r^T \tag{3}$$
where $\boldsymbol{e}$ denotes the all-ones column vector (check this!)
- ERO noise model
$$\mathbb{E}C_{ij} = (r_i - r_j)(1 - \eta)p, \tag{4}$$
- $\mathbb{E}C$ is a rank-2 skew-symmetric matrix
$$\mathbb{E}C = (1 - \eta)p(r\boldsymbol{e}^T - \boldsymbol{e}r^T) \tag{5}$$
- decompose the given data matrix C as
$$C = \mathbb{E}C + R \tag{6}$$

Singular Value Decomposition (SVD) ranking

- for cardinal measurements $C_{ij} = r_i - r_j$, the noiseless matrix of rank offsets $C = (C_{ij})_{1 \le i,j \le n}$ is a skew-symmetric of even rank 2
$$C = r\boldsymbol{e}^T - \boldsymbol{e}r^T \tag{3}$$
where $\boldsymbol{e}$ denotes the all-ones column vector (check this!)
- ERO noise model
$$\mathbb{E}C_{ij} = (r_i - r_j)(1 - \eta)p, \tag{4}$$
- $\mathbb{E}C$ is a rank-2 skew-symmetric matrix
$$\mathbb{E}C = (1 - \eta)p(r\boldsymbol{e}^T - \boldsymbol{e}r^T) \tag{5}$$
- decompose the given data matrix C as
$$C = \mathbb{E}C + R \tag{6}$$
- can recover the ordering from top 2 singular vectors of C: $\{v_1, v_2, -v_1, -v_2\}$; order their entries, infer rankings, and choose whichever minimizes the number of upsets

Singular Value Decomposition (SVD) ranking

- for cardinal measurements $C_{ij} = r_i - r_j$, the noiseless matrix of rank offsets $C = (C_{ij})_{1 \leq i,j \leq n}$ is a skew-symmetric of even rank 2
$$C = r\boldsymbol{e}^T - \boldsymbol{e}r^T \tag{3}$$
where $\boldsymbol{e}$ denotes the all-ones column vector (check this!)
- ERO noise model
$$\mathbb{E}C_{ij} = (r_i - r_j)(1 - \eta)p, \tag{4}$$
- $\mathbb{E}C$ is a rank-2 skew-symmetric matrix
$$\mathbb{E}C = (1 - \eta)p(r\boldsymbol{e}^T - \boldsymbol{e}r^T) \tag{5}$$
- decompose the given data matrix C as
$$C = \mathbb{E}C + R \tag{6}$$
- can recover the ordering from top 2 singular vectors of C: $\{v_1, v_2, -v_1, -v_2\}$; order their entries, infer rankings, and choose whichever minimizes the number of upsets
- amenable to a theoretical analysis using tools from random matrix theory on rank-2 deformations of random matrices.

Ranking via Least-Squares

- $m = |E(G)|$
- recall that instead of clean measurements

$$T_{ij} = r_i - r_j \tag{7}$$

Ranking via Least-Squares

- $m = |E(G)|$
- recall that instead of clean measurements
$$T_{ij} = r_i - r_j \tag{7}$$
we observe noisy measurements
$$\underbrace{T_{ij} + \text{noise}}_{C_{ij}} = r_i - r_j \tag{8}$$

Ranking via Least-Squares

- $m = |E(G)|$
- recall that instead of clean measurements

$$T_{ij} = r_i - r_j \tag{7}$$

we observe noisy measurements

$$\underbrace{T_{ij} + \text{noise}}_{C_{ij}} = r_i - r_j \tag{8}$$

thus we can set up a linear system of equations

$$r_i - r_j \approx C_{ij} \tag{9}$$

Ranking via Least-Squares

- $m = |E(G)|$
- recall that instead of clean measurements

$$T_{ij} = r_i - r_j \tag{7}$$

we observe noisy measurements

$$\underbrace{T_{ij} + \text{noise}}_{C_{ij}} = r_i - r_j \tag{8}$$

thus we can set up a linear system of equations

$$r_i - r_j \approx C_{ij} \tag{9}$$

and solve this in the least squares sense.

Ranking via Least-Squares

- $m = |E(G)|$
- recall that instead of clean measurements

$$T_{ij} = r_i - r_j \tag{7}$$

  we observe noisy measurements

$$\underbrace{T_{ij} + \text{noise}}_{C_{ij}} = r_i - r_j \tag{8}$$

  thus we can set up a linear system of equations

$$r_i - r_j \approx C_{ij} \tag{9}$$

  and solve this in the least squares sense.
- denote by B the edge-vertex incidence matrix of size $m \times n$

$$B_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E(G), \quad \text{and } i > j \\ -1 & \text{if } (i,j) \in E(G), \quad \text{and } i < j \\ 0 & \text{if } (i,j) \notin E(G) \end{cases} \tag{10}$$

Ranking via Least-Squares

- $m = |E(G)|$
- recall that instead of clean measurements

$$T_{ij} = r_i - r_j \tag{7}$$

we observe noisy measurements

$$\underbrace{T_{ij} + \textcolor{red}{\text{noise}}}_{C_{ij}} = r_i - r_j \tag{8}$$

thus we can set up a linear system of equations

$$r_i - r_j \approx C_{ij} \tag{9}$$

and solve this in the least squares sense.

- denote by B the edge-vertex incidence matrix of size $m \times n$

$$B_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E(G), \quad \text{and } i > j \\ -1 & \text{if } (i,j) \in E(G), \quad \text{and } i < j \\ 0 & \text{if } (i,j) \notin E(G) \end{cases} \tag{10}$$

- w the vector of size $m \times 1$ containing all pairwise comparisons $w(e) = C_{ij}$, for all edges $e = (i,j) \in E(G)$

Ranking via Least-Squares

- $m = |E(G)|$
- recall that instead of clean measurements
$$T_{ij} = r_i - r_j \tag{7}$$
we observe noisy measurements
$$\underbrace{T_{ij} + \text{noise}}_{C_{ij}} = r_i - r_j \tag{8}$$
thus we can set up a linear system of equations
$$r_i - r_j \approx C_{ij} \tag{9}$$
and solve this in the least squares sense.
- denote by B the edge-vertex incidence matrix of size $m \times n$
$$B_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E(G), \quad \text{and } i > j \\ -1 & \text{if } (i,j) \in E(G), \quad \text{and } i < j \\ 0 & \text{if } (i,j) \notin E(G) \end{cases} \tag{10}$$
- w the vector of size $m \times 1$ containing all pairwise comparisons $w(e) = C_{ij}$, for all edges $e = (i,j) \in E(G)$
- least-squares solution to the ranking problem
$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad ||Bx - w||_2^2. \tag{11}$$

Serial-Rank (NIPS 2014; JMLR 2016)

$$C_{ij} = \begin{cases} 1 & \text{if i is ranked higher than j} \\ 0 & \text{if i and j are tied, or comparison is not available} \\ -1 & \text{if j is ranked higher than i} \end{cases} \tag{12}$$

- the pairwise similarity matrix is given by

$$S_{ij}^{match} = \sum_{k=1}^{n} \left(\frac{1 + C_{ik} C_{jk}}{2} \right) \tag{13}$$

Serial-Rank (NIPS 2014; JMLR 2016)

$$C_{ij} = \begin{cases} 1 & \text{if i is ranked higher than j} \\ 0 & \text{if i and j are tied, or comparison is not available} \\ -1 & \text{if j is ranked higher than i} \end{cases} \tag{12}$$

- the pairwise similarity matrix is given by

$$S_{ij}^{match} = \sum_{k=1}^{n} \left(\frac{1 + C_{ik} C_{jk}}{2} \right) \tag{13}$$

- $C_{ik} C_{jk} = 1$ whenever *i* and *j* have the same signs, and $C_{ik} C_{jk} = -1$ whenever they have opposite signs
- S_{ij}^{match} counts the number of matching comparisons between *i* and *j* with a third reference item *k*

Serial-Rank (NIPS 2014; JMLR 2016)

$$C_{ij} = \begin{cases} 1 & \text{if i is ranked higher than j} \\ 0 & \text{if i and j are tied, or comparison is not available} \\ -1 & \text{if j is ranked higher than i} \end{cases} \quad (12)$$

- the pairwise similarity matrix is given by

$$S_{ij}^{match} = \sum_{k=1}^{n} \left(\frac{1 + C_{ik}C_{jk}}{2} \right) \quad (13)$$

- $C_{ik}C_{jk} = 1$ whenever i and j have the same signs, and $C_{ik}C_{jk} = -1$ whenever they have opposite signs
- S_{ij}^{match} counts the number of matching comparisons between i and j with a third reference item k
- intuition: players that beat the same players and are beaten by the same players should have a similar ranking in the final solution

Serial-Rank (NIPS 2014; JMLR 2016)

$$C_{ij} = \begin{cases} 1 & \text{if i is ranked higher than j} \\ 0 & \text{if i and j are tied, or comparison is not available} \\ -1 & \text{if j is ranked higher than i} \end{cases} \tag{12}$$

- the pairwise similarity matrix is given by

$$S_{ij}^{match} = \sum_{k=1}^{n} \left(\frac{1 + C_{ik} C_{jk}}{2} \right) \tag{13}$$

- $C_{ik} C_{jk} = 1$ whenever i and j have the same signs, and $C_{ik} C_{jk} = -1$ whenever they have opposite signs
- S_{ij}^{match} counts the number of matching comparisons between i and j with a third reference item k
- intuition: players that beat the same players and are beaten by the same players should have a similar ranking in the final solution
- final similarity matrix is given by

$$S^{match} = \frac{1}{2} \left(n\mathbf{1}\mathbf{1}^T + CC^T \right) \tag{14}$$

Algorithm 1 Serial-Rank: an algorithm for spectral ranking using seriation, proposed by Fogel, d'Aspremont and Vojnovic

Require: A set of pairwise comparisons $C_{ij} \in \{-1, 0, 1\}$ or [-1,1]

1: Compute a similarity matrix as shown in (13)

2: Compute the associated graph Laplacian matrix

$$L_S = D - S \tag{15}$$

where D is a diagonal matrix $D = \textbf{diag}\,(S\mathbf{1})$, i.e., $D_{ii} = \sum_{j=1}^{n} S_{i,j}$ is the degree of node i.

3: Compute the Fiedler vector of S (eigenvector corresponding to the smallest nonzero eigenvalue of L_S).

4: Output the ranking induced by sorting the Fiedler vector of S, with the global ordering (increasing or decreasing order) chosen such that the number of upsets is minimized.

Ranking via Group Synchronization

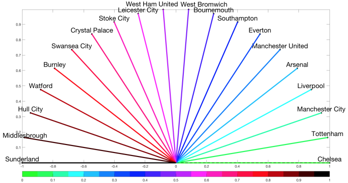
Recover $r_1, \ldots, r_n \in \mathbb{R}$ from a sparse noisy set of pairwise comparisons

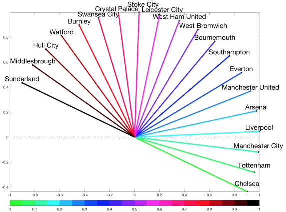
$$C_{ij} = r_i - r_j + \text{Noise} \quad (i,j) \in E(G)$$

Ranking via Group Synchronization

Recover $r_1, \ldots, r_n \in \mathbb{R}$ from a sparse noisy set of pairwise comparisons

$$C_{ij} = r_i - r_j + \text{Noise} \quad (i,j) \in E(G)$$





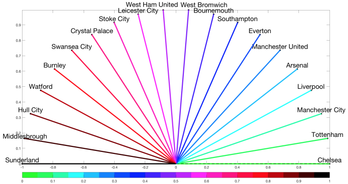
Map all rank offsets C_{ij} to an angle $\Theta_{ij} \in [0, 2\pi\delta)$

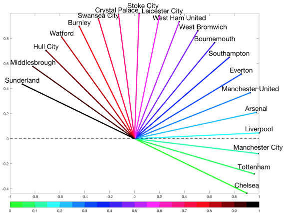
$$\Theta_{ij} := 2\pi\delta \frac{C_{ij}}{n-1} \qquad (16)$$

Ranking via Group Synchronization

Recover $r_1, \ldots, r_n \in \mathbb{R}$ from a sparse noisy set of pairwise comparisons

$$C_{ij} = r_i - r_j + \text{Noise} \quad (i,j) \in E(G)$$





Map all rank offsets C_{ij} to an angle $\Theta_{ij} \in [0, 2\pi\delta)$

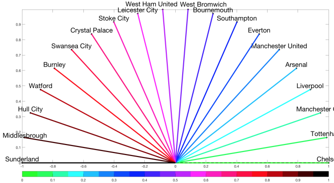
$$\Theta_{ij} := 2\pi\delta \frac{C_{ij}}{n-1} \qquad (16)$$

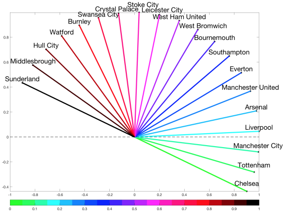
$$H_{ij} = \begin{cases} e^{\imath\Theta_{ij}} & \text{if } (i,j) \in E \\ 0 & \text{if } (i,j) \notin E. \end{cases} \qquad (17)$$

Ranking via Group Synchronization

Recover $r_1, \ldots, r_n \in \mathbb{R}$ from a sparse noisy set of pairwise comparisons

$$C_{ij} = r_i - r_j + \text{Noise} \quad (i,j) \in E(G)$$





Map all rank offsets C_{ij} to an angle $\Theta_{ij} \in [0, 2\pi\delta)$

$$\Theta_{ij} := 2\pi\delta \frac{C_{ij}}{n-1} \qquad (16)$$

$$H_{ij} = \begin{cases} e^{\imath\Theta_{ij}} & \text{if } (i,j) \in E \\ 0 & \text{if } (i,j) \notin E. \end{cases} \qquad (17)$$

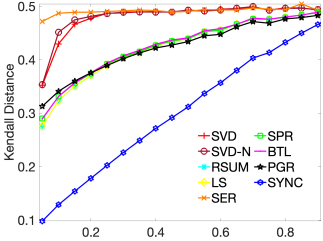


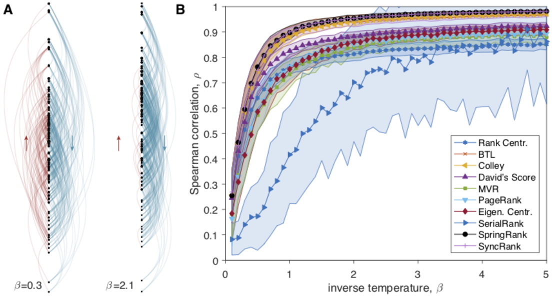
Figure: $p = 0.05$, $n \stackrel{\gamma}{=} 1000$ (Gamma)

$$\underset{\theta_1,\ldots,\theta_n \in [0,2\pi)}{\text{maximize}} \sum_{i,j=1}^{n} e^{-\iota\theta_i} H_{ij} e^{\iota\theta_j} \quad + \quad \text{relax (spectral or SDP)} \qquad (18)$$

A physical model for efficient ranking in networks

Caterina De Bacco,[1,2,*] Daniel B. Larremore,[3,4,2,†] and Cristopher Moore[2,‡]
[1] Data Science Institute, Columbia University, New York, NY 10027, USA
[2] Santa Fe Institute, Santa Fe, NM 87501, USA
[3] Department of Computer Science, University of Colorado, Boulder, CO 80309, USA
[4] BioFrontiers Institute, University of Colorado, Boulder, CO 80303, USA

We present a physically-inspired model and an efficient algorithm to infer hierarchical rankings of nodes in directed networks. It assigns real-valued ranks to nodes rather than simply ordinal ranks, and it formalizes the assumption that interactions are more likely to occur between individuals with similar ranks. It provides a natural statistical significance test for the inferred hierarchy, and it can be used to perform inference tasks such as predicting the existence or direction of edges. The ranking is obtained by solving a linear system of equations, which is sparse if the network is; thus the resulting algorithm is extremely efficient and scalable. We illustrate these findings by analyzing real and synthetic data, including datasets from animal behavior, faculty hiring, social support networks, and sports tournaments. We show that our method often outperforms a variety of others, in both speed and accuracy, in recovering the underlying ranks and predicting edge directions.



- C. De Bacco, D. B. Larremore and C. Moore, A physical model for efficient ranking in networks, Science Advances (2018). + Implementations in Python, Matlab, R.

Recovering planted partial rankings

- V is the set of all players
- $V = V_\alpha \cup V_\beta$
- α-players
- β-players

Measurement between

- two α-players is clean or less noisy ($\eta_1 = 0$)
- two β-players is very noisy ($\eta_2 = 1$)
- an α-player and a β-player is very noisy ($\eta_2 = 1$)

Recovering planted partial rankings

- V is the set of all players
- $V = V_\alpha \cup V_\beta$
- α-players
- β-players

Measurement between

- two α-players is clean or less noisy ($\eta_1 = 0$)
- two β-players is very noisy ($\eta_2 = 1$)
- an α-player and a β-player is very noisy ($\eta_2 = 1$)

Consider the ensemble: $\mathcal{G}(n = 250, \alpha = 0.3, \eta_1 = 0, \eta_2 = 1)$

Recovering planted partial rankings

- V is the set of all players
- $V = V_\alpha \cup V_\beta$
- α-players
- β-players

Measurement between

- two α-players is clean or less noisy ($\eta_1 = 0$)
- two β-players is very noisy ($\eta_2 = 1$)
- an α-player and a β-player is very noisy ($\eta_2 = 1$)

Consider the ensemble: $\mathcal{G}(n = 250, \alpha = 0.3, \eta_1 = 0, \eta_2 = 1)$
(α is the fraction of α-players. $|V_\alpha| = \alpha n$)

Question: can you recover V_α and V_β ?

Recovering planted partial rankings

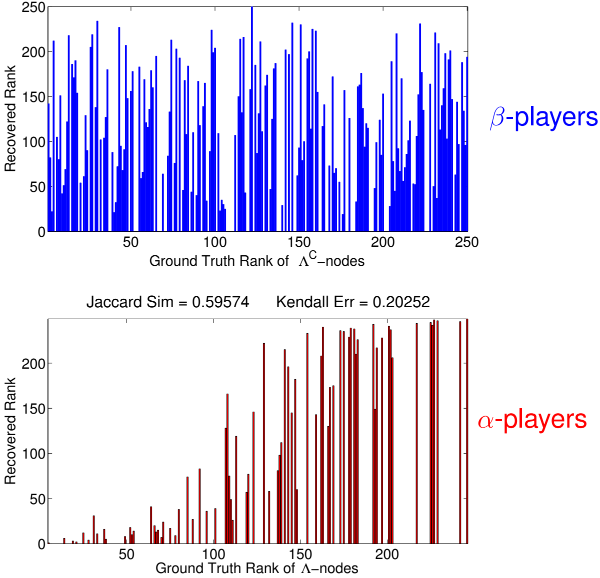
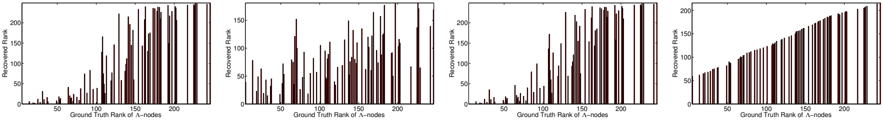


Figure: (Least-squares ranking) The top (respectively, bottom) subplot corresponds to the recovered rankings of the β-players, respectively the α-players.

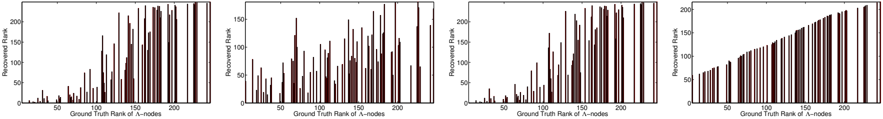
Recovering planted partial rankings



(a) LS (b) SER (c) RC (d) SYNC

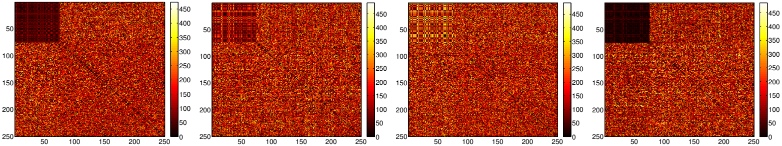
Figure: The plots corresponds to the recovered rankings of the α-players.

Recovering planted partial rankings



(a) LS (b) SER (c) RC (d) SYNC

Figure: The plots corresponds to the recovered rankings of the α-players.



(a) LS (b) SER (c) RC (d) SYNC

Figure: Residual matrices R with $R_{ij} = |C_{ij} - \widehat{C}_{ij}|$, and $\widehat{C}$ is the denoised comparison matrix $\widehat{C}_{ij} = \hat{r}_i - \hat{r}_j$. Here $\mathcal{G}(n = 250, \beta = 0.3, \eta_1 = 0, \eta_2 = 1)$. The top left block diagonal sub-matrix corresponds to the residual between (α, α) matches; while the rest of the sub-matrices corresponds to (β, β) (bottom right block diagonal) and (α, β).

Price Impact of Order Flow Imbalances: Multi-level, Cross-asset and Forecasting

August 10, 2022

Joint work with Rama Cont, Chao Zhang

3

Outline

Introduction

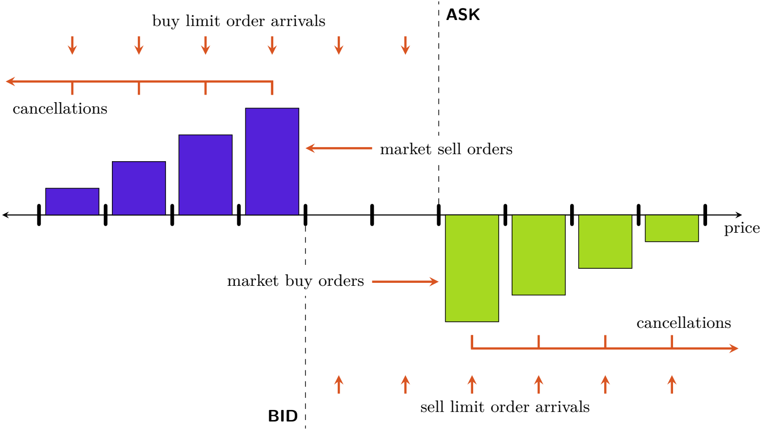
Variables & Data

Contemporaneous Impact

Forecasting Impact

Summary

Limit Order Book (LOB)



Source: `https://help.quantower.com`

Each unique price corresponds to a level in the limit order book. Market orders are immediately executed. For limit orders, market participants can post a new order, cancel an order, or partially cancel (decrease order size), or modify (increase order size).

Price Impact

What is price impact?

- price impact refers to the relation between order flow and price change Eisler et al. (2012)
- *"a buy trade should push the price up"* Bouchaud (2010)

Price Impact

What is price impact?

- price impact refers to the relation between order flow and price change Eisler et al. (2012)
- *”a buy trade should push the price up"* Bouchaud (2010)

Why do we study price impact?

- For traders, *price impact is tantamount to a cost: their second buy trade is on average more expensive than the first because of their own impac* Bouchaud (2010)
- *the optimal liquidation of a large block of shares, given a fixed time horizon* Cont et al. (2014)
- *a fundamental mechanism of price formation* Cont et al. (2014)

Cross Impact

What is cross impact?

- The price of a given asset might be moved by the orders of other assets Capponi and Cont (2020)

Cross Impact

What is cross impact?

- The price of a given asset might be moved by the orders of other assets Capponi and Cont (2020)

Why do we study cross impact?

- Cross impact is important in accurate assessment of portfolio trading cost.
- Cross impact has been much less studied and its evidences are mixed.
- What information is transmitted across assets?

Outline

Introduction

Variables & Data

Contemporaneous Impact

Forecasting Impact

Summary

Variables

Best-level OFI Cont et al. (2014)

$$\mathbf{OFI}_{i,t}^{\mathbf{1},(h)} := \sum_{n=N(t-h)+\mathbf{1}}^{N(t)} q_{i,n}^{\mathbf{1},b} \mathbf{1}_{\{P_{i,n}^{\mathbf{1},b} \geq P_{i,n-\mathbf{1}}^{\mathbf{1},b}\}} - q_{i,n-\mathbf{1}}^{\mathbf{1},b} \mathbf{1}_{\{P_{i,n}^{\mathbf{1},b} \leq P_{i,n-\mathbf{1}}^{\mathbf{1},b}\}} - q_{i,n}^{\mathbf{1},s} \mathbf{1}_{\{P_{i,n}^{\mathbf{1},s} \leq P_{i,n-\mathbf{1}}^{\mathbf{1},s}\}} + q_{i,n-\mathbf{1}}^{\mathbf{1},s} \mathbf{1}_{\{P_{i,n}^{\mathbf{1},s} \geq P_{i,n-\mathbf{1}}^{\mathbf{1},s}\}} \quad \mathbf{(1)}$$

- $P_{i,n}^{1,b}$ and $q_{i,n}^{1,b}$ denote the best bid price and size (in # of shares) of stock i, resp.
- $P_{i,n}^{1,s}$ and $q_{i,n}^{1,s}$ denote the ask price and ask size at the best level, respectively.
- We enumerate the observations of the bid and the ask by n, and $N(t-h)+1$ and $N(h)$ are the index of the first and last order book event in the interval $(t-h, t]$.
- OFI increases when: (i) the bid size increases and the best bid remains the same; (ii) the ask size decreases and the best ask price remains the same; (iii) the best bid/ask price increases.

Variables

Best-level OFI Cont et al. (2014)

$$\mathbf{OFI}_{i,t}^{1,(h)} := \sum_{n=N(t-h)+1}^{N(t)} q_{i,n}^{1,b} \mathbf{1}_{\{P_{i,n}^{1,b} \geq P_{i,n-1}^{1,b}\}} - q_{i,n-1}^{1,b} \mathbf{1}_{\{P_{i,n}^{1,b} \leq P_{i,n-1}^{1,b}\}} - q_{i,n}^{1,s} \mathbf{1}_{\{P_{i,n}^{1,s} \leq P_{i,n-1}^{1,s}\}} + q_{i,n-1}^{1,s} \mathbf{1}_{\{P_{i,n}^{1,s} \geq P_{i,n-1}^{1,s}\}} \tag{1}$$

- $P_{i,n}^{1,b}$ and $q_{i,n}^{1,b}$ denote the best bid price and size (in # of shares) of stock i, resp.
- $P_{i,n}^{1,s}$ and $q_{i,n}^{1,s}$ denote the ask price and ask size at the best level, respectively.
- We enumerate the observations of the bid and the ask by n, and $N(t-h)+1$ and $N(h)$ are the index of the first and last order book event in the interval $(t-h, t]$.
- OFI increases when: (i) the bid size increases and the best bid remains the same; (ii) the ask size decreases and the best ask price remains the same; (iii) the best bid/ask price increases.

Deeper-level OFI Cont et al. (2014); Xu et al. (2018)

$$\mathbf{OFI}_{i,t}^{m,(h)} := \sum_{n=N(t-h)+1}^{N(t)} q_{i,n}^{m,b} \mathbf{1}_{\{P_{i,n}^{m,b} \geq P_{i,n-1}^{m,b}\}} - q_{i,n-1}^{m,b} \mathbf{1}_{\{P_{i,n}^{m,b} \leq P_{i,n-1}^{m,b}\}} - q_{i,n}^{m,s} \mathbf{1}_{\{P_{i,n}^{m,s} \leq P_{i,n-1}^{m,s}\}} + q_{i,n-1}^{m,s} \mathbf{1}_{\{P_{i,n}^{m,s} \geq P_{i,n-1}^{m,s}\}}. \tag{2}$$

- $P_{i,n}^{m,b}$ and $q_{i,n}^{m,b}$ denote, respectively, the bid price and bid size (in number of shares) at level m. Similarly, $P_{i,n}^{m,s}$ and $q_{i,n}^{m,s}$ denote the ask price and ask size at level m.

Variables

Best-level OFI Cont et al. (2014)

$$\mathrm{OFI}_{i,t}^{1,(h)} := \sum_{n=N(t-h)+1}^{N(t)} q_{i,n}^{1,b} \mathbf{1}_{\{P_{i,n}^{1,b} \geq P_{i,n-1}^{1,b}\}} - q_{i,n-1}^{1,b} \mathbf{1}_{\{P_{i,n}^{1,b} \leq P_{i,n-1}^{1,b}\}} - q_{i,n}^{1,s} \mathbf{1}_{\{P_{i,n}^{1,s} \leq P_{i,n-1}^{1,s}\}} + q_{i,n-1}^{1,s} \mathbf{1}_{\{P_{i,n}^{1,s} \geq P_{i,n-1}^{1,s}\}} \quad (1)$$

- $P_{i,n}^{1,b}$ and $q_{i,n}^{1,b}$ denote the best bid price and size (in # of shares) of stock i, resp.
- $P_{i,n}^{1,s}$ and $q_{i,n}^{1,s}$ denote the ask price and ask size at the best level, respectively.
- We enumerate the observations of the bid and the ask by n, and $N(t-h)+1$ and $N(h)$ are the index of the first and last order book event in the interval $(t-h, t]$.
- OFI increases when: (i) the bid size increases and the best bid remains the same; (ii) the ask size decreases and the best ask price remains the same; (iii) the best bid/ask price increases.

Deeper-level OFI Cont et al. (2014); Xu et al. (2018)

$$\mathrm{OFI}_{i,t}^{m,(h)} := \sum_{n=N(t-h)+1}^{N(t)} q_{i,n}^{m,b} \mathbf{1}_{\{P_{i,n}^{m,b} \geq P_{i,n-1}^{m,b}\}} - q_{i,n-1}^{m,b} \mathbf{1}_{\{P_{i,n}^{m,b} \leq P_{i,n-1}^{m,b}\}} - q_{i,n}^{m,s} \mathbf{1}_{\{P_{i,n}^{m,s} \leq P_{i,n-1}^{m,s}\}} + q_{i,n-1}^{m,s} \mathbf{1}_{\{P_{i,n}^{m,s} \geq P_{i,n-1}^{m,s}\}}. \quad (2)$$

- $P_{i,n}^{m,b}$ and $q_{i,n}^{m,b}$ denote, respectively, the bid price and bid size (in number of shares) at level m. Similarly, $P_{i,n}^{m,s}$ and $q_{i,n}^{m,s}$ denote the ask price and ask size at level m.

Logarithmic return $r_{i,t}^{(h)} = \log\left(\frac{P_{i,t}}{P_{i,t-h}}\right).$

Data

- We use the Nasdaq ITCH data from LOBSTER to compute OFIs and returns during the intraday time interval 10:00AM-3:30PM.

Data

- We use the Nasdaq ITCH data from LOBSTER to compute OFIs and returns during the intraday time interval 10:00AM-3:30PM.

- The reason for excluding the first and last 30 minutes of the trading day is due to the increased volatility near the opening and closing session.

Data

- We use the Nasdaq ITCH data from LOBSTER to compute OFIs and returns during the intraday time interval 10:00AM-3:30PM.
- The reason for excluding the first and last 30 minutes of the trading day is due to the increased volatility near the opening and closing session.
- Our data includes the top 100 components of S&P500 index, for the period 2017-01-01 to 2019-12-31 (ongoing updated study with all 500 components).

message file.

Time (sec)	Event Type	Order ID	Size	Price	Direction
⋮	⋮	⋮	⋮	⋮	⋮
34713.685155243	1	206833312	100	118600	-1
34714.133632201	3	206833312	100	118600	-1
⋮	⋮	⋮	⋮	⋮	⋮

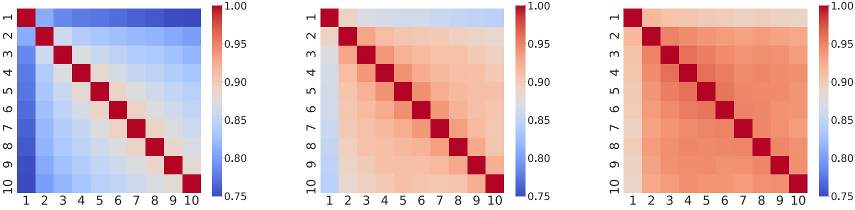
order book file.

Ask Price 1	Ask Size 1	Bid Price 1	Bid Size 1	Ask Price 2	Ask Size 2	Bid Price 2	Bid Size 2	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1186600	9484	118500	8800	118700	22700	118400	14930	...
1186600	9384	118500	8800	118700	22700	118400	14930	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

variable explanation.

- Ask Price 1: Level 1 ask price (best ask price)
- Ask Size 1: Level 1 ask volume (best ask volume)
- Bid Price 1: Level 1 bid price (best bid price)
- Bid Size 1: Level 1 bid volume (best bid volume)

Correlation matrix of multi-level OFIs



(a) Average (b) AAPL (c) JPM

Figure 1: Correlation matrix of multi-level OFIs. (a) averaged across stocks, (b)-(c): correlation matrix of Apple (AAPL) and JPMorgan Chase (JPM). The x-axis and y-axis represent different levels of OFIs.

- There exist strong relationships among multi-level OFIs.
- Best-level OFI exhibits the smallest correlation with any of the remaining nine levels, a pattern which persists across the different stocks.

Principal component of multi-level OFIs

Principal Components Analysis (PCA) is a widely-used statistical procedure that applies an orthogonal transformation to convert a number of correlated variables into a smaller number of uncorrelated variables, i.e. *principal components* (PCs).

Principal Component	1	2	3	4	5	6	7	8	9	10
Explained Variance Ratio (%)	83.71	6.74	3.25	1.98	1.33	0.97	0.73	0.56	0.42	0.31

Table 1: Average percentage of variance attributed to each PC (i.e., the ratio between the variance of each PC and the total variance).

Principal component of multi-level OFIs

Principal Components Analysis (PCA) is a widely-used statistical procedure that applies an orthogonal transformation to convert a number of correlated variables into a smaller number of uncorrelated variables, i.e. *principal components* (PCs).

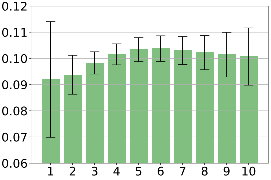
Principal Component	1	2	3	4	5	6	7	8	9	10
Explained Variance Ratio (%)	83.71	6.74	3.25	1.98	1.33	0.97	0.73	0.56	0.42	0.31

Table 1: Average percentage of variance attributed to each PC (i.e., the ratio between the variance of each PC and the total variance).

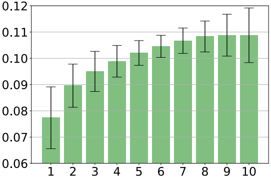
Integrated OFI. Denoted by $\mathbf{ofi}_{i,t}^{(h)} = \left(\mathrm{ofi}_{i,t}^{1,(h)}, \cdots, \mathrm{ofi}_{i,t}^{10,(h)}\right)^{\top}$. We apply the first principal component to transform the multi-level OFI vector $\mathbf{ofi}_{i,t}^{(h)}$ to $\mathrm{ofi}_{i,t}^{I,(h)}$, denoted as integrated OFI. Let $\boldsymbol{w}_1$ be the first principal component

$$\mathrm{ofi}_{i,t}^{I,(h)} = \frac{\boldsymbol{w}_1^T \mathbf{ofi}_{i,t}^{(h)}}{\|\boldsymbol{w}_1\|_1}. \tag{3}$$

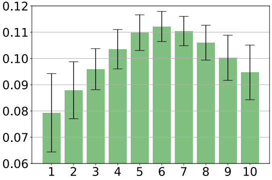
First principal component of multi-level OFIs



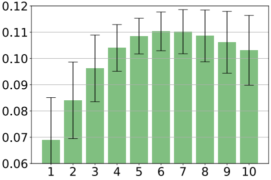
(a) Average



(b) AAPL



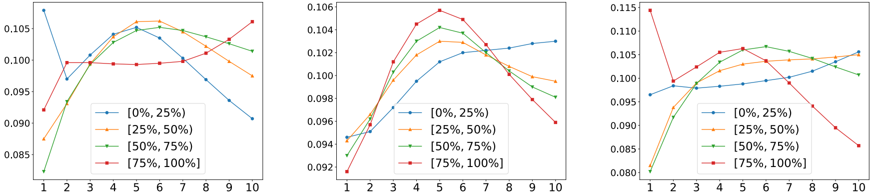
(c) JPM



(d) JNJ

Figure 2: First PC of multi-level OFIs. (a) averaged across stocks, (b)-(d): first PC of Apple (AAPL), JPMorgan Chase (JPM), and Johnson & Johnson (JNJ). The x-axis represents different levels of OFIs and y-axis represents the weights of OFIs in the first PC.

First principal component of multi-level OFIs



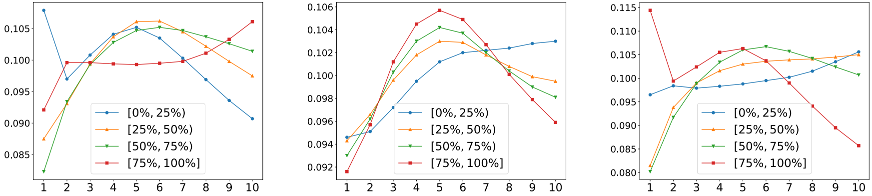
(a) Volume (b) Volatility (c) Spread

Figure 3: First PC of multi-level OFIs, in quantile buckets for various stock characteristics. The *x*-axis indexes the top 10 levels of the OFIs.

- **Volume**: trading volume on the previous trading day.
- **Volatility**: volatility of one-minute returns during the previous trading day.
- **Spread**: average bid-ask spread during the previous trading day.

$[0\%, 25\%)$, respectively $[75\%, 100\%]$, denote the subset of stocks with the lowest, respectively highest, 25% values for a given stock characteristic.

First principal component of multi-level OFIs



(a) Volume (b) Volatility (c) Spread

Figure 3: First PC of multi-level OFIs, in quantile buckets for various stock characteristics. The x-axis indexes the top 10 levels of the OFIs.

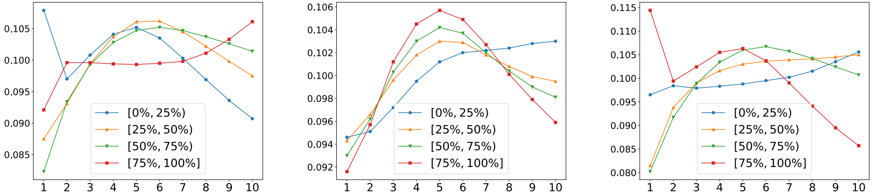
- **Volume**: trading volume on the previous trading day.
- **Volatility**: volatility of one-minute returns during the previous trading day.
- **Spread**: average bid-ask spread during the previous trading day.

$[0\%, 25\%)$, respectively $[75\%, 100\%]$, denote the subset of stocks with the lowest, respectively highest, 25% values for a given stock characteristic.

Surprising pattern

- for *high-volume*, and *low-volatility stocks*, OFIs deeper in the LOB receive more weight in the PC1

First principal component of multi-level OFIs



(a) Volume (b) Volatility (c) Spread

Figure 3: First PC of multi-level OFIs, in quantile buckets for various stock characteristics. The *x*-axis indexes the top 10 levels of the OFIs.

- **Volume**: trading volume on the previous trading day.
- **Volatility**: volatility of one-minute returns during the previous trading day.
- **Spread**: average bid-ask spread during the previous trading day.

$[0\%, 25\%)$, respectively $[75\%, 100\%]$, denote the subset of stocks with the lowest, respectively highest, 25% values for a given stock characteristic.

Surprising pattern

- for *high-volume*, and *low-volatility stocks*, OFIs deeper in the LOB receive more weight in the PC1
- for *low-volume*, and *large-spread stocks*, the best-level OFIs account more than the deeper-level OFIs.

Outline

Contemporaneous Models

Price Impact based on best-level OFIs, estimated via OLS

$$\mathbf{PI}^{[1]}: \quad r_{i,t}^{(h)} = \alpha_i^{[1]} + \beta_i^{[1]} \mathrm{ofi}_{i,t}^{1,(h)} + \epsilon_{i,t}^{[1]}. \tag{4}$$

Price Impact based on integrated OFIs, estimated via OLS

$$\mathbf{PI}^{I}: \quad r_{i,t}^{(h)} = \alpha_i^{I} + \beta_i^{I} \mathrm{ofi}_{i,t}^{I} + \epsilon_{i,t}^{I}. \tag{5}$$

Contemporaneous Models

Price Impact based on best-level OFIs, estimated via OLS

$$\mathbf{PI}^{[1]}: \quad r_{i,t}^{(h)} = \alpha_i^{[1]} + \beta_i^{[1]} \mathrm{ofi}_{i,t}^{1,(h)} + \epsilon_{i,t}^{[1]}. \tag{4}$$

Price Impact based on integrated OFIs, estimated via OLS

$$\mathbf{PI}^{I}: \quad r_{i,t}^{(h)} = \alpha_i^{I} + \beta_i^{I} \mathrm{ofi}_{i,t}^{I} + \epsilon_{i,t}^{I}. \tag{5}$$

Cross Impact based on best-level OFIs, estimated via LASSO

$$\mathbf{CI}^{[1]}: \quad r_{i,t}^{(h)} = \alpha_i^{[1]} + \beta_{i,i}^{[1]} \mathrm{ofi}_{i,t}^{1,(h)} + \sum_{j \neq i} \beta_{i,j}^{[1]} \mathrm{ofi}_{j,t}^{1,(h)} + \eta_{i,t}^{[1]}. \tag{6}$$

Cross Impact based on integrated OFIs, estimated via LASSO

$$\mathbf{CI}^{I}: \quad r_{i,t}^{(h)} = \alpha_i^{I} + \beta_{i,i}^{I} \mathrm{ofi}_{i,t}^{I,(h)} + \sum_{j \neq i} \beta_{i,j}^{I} \mathrm{ofi}_{j,t}^{I,(h)} + \eta_{i,t}^{I}. \tag{7}$$

Sparsity of Cross-Impact terms

- Ordinary least squares (OLS) regression becomes ill-posed when observations are fewer than parameters.

Sparsity of Cross-Impact terms

- Ordinary least squares (OLS) regression becomes ill-posed when observations are fewer than parameters.

- Capponi and Cont (2020) find that a certain number of cross-impact coefficients $\beta_{i,j}(j \neq i)$ from their OLS regressions are not statistically significant at the 1% significance level.

Sparsity of Cross-Impact terms

- Ordinary least squares (OLS) regression becomes ill-posed when observations are fewer than parameters.
- Capponi and Cont (2020) find that a certain number of cross-impact coefficients $\beta_{i,j} (j \neq i)$ from their OLS regressions are not statistically significant at the 1% significance level.
- Least Absolute Shrinkage and Selection Operator (LASSO) is a regression method that performs both variable selection and regularization, in order to enhance the prediction accuracy and interpretability of regression models.

Sparsity of Cross-Impact terms

- Ordinary least squares (OLS) regression becomes ill-posed when observations are fewer than parameters.
- Capponi and Cont (2020) find that a certain number of cross-impact coefficients $\beta_{i,j} (j \neq i)$ from their OLS regressions are not statistically significant at the 1% significance level.
- Least Absolute Shrinkage and Selection Operator (LASSO) is a regression method that performs both variable selection and regularization, in order to enhance the prediction accuracy and interpretability of regression models.

In the cross-impact model based on best-level OFIs, parameter $\beta_{i,j}^{[1]}$ $(j \neq i)$ represents the influence of the j-th stock's OFI on the return of stock i, after accounting for the own best-level OFI impact of stock i.

Comparison of Performance

Implementation. We use a 30-minute estimation window and within each window, data associated with returns and OFIs are computed for every 10 seconds. Altogether, this amounts to $30 \times 60 / 10 = 180$ observations for each regression.

17

Comparison of Performance

Implementation. We use a 30-minute estimation window and within each window, data associated with returns and OFIs are computed for every 10 seconds. Altogether, this amounts to $30 \times 60 \,/\, 10 = 180$ observations for each regression.

Performance evaluation.

- In-Sample R^2, i.e. adjusted-R^2 on training samples.
- Out-Of-Sample R^2. We use the fitted model to estimate returns on the following 30-minute data and compute the corresponding R^2.

Comparison of Performance

Implementation. We use a 30-minute estimation window and within each window, data associated with returns and OFIs are computed for every 10 seconds. Altogether, this amounts to $30 \times 60 / 10 = 180$ observations for each regression.

Performance evaluation.

- In-Sample R^2, i.e. adjusted-R^2 on training samples.
- Out-Of-Sample R^2. We use the fitted model to estimate returns on the following 30-minute data and compute the corresponding R^2.

	Best-level		Integrated	
	$\mathbf{PI}^{[1]}$	$\mathbf{CI}^{[1]}$	$\mathbf{PI}^{I}$	$\mathbf{CI}^{I}$
In-Sample (%)	71.16	73.87	87.14	**87.85**
Out-Of-Sample (%)	64.64	66.03	**83.83**	83.62

Table 2: Statistical performance of the contemporaneous models, namely $\mathbf{PI}^{[1]}$, $\mathbf{PI}^{I}$, $\mathbf{CI}^{[1]}$, and $\mathbf{CI}^{I}$.

• No need to introduce cross-impact terms for modeling the contemporaneous returns, as long as the OFI incorporates information from multiple levels, not just the best level.

Coefficients

The matrix of coefficients of the integrated OFIs has relatively smaller singular values and a faster decay of the spectrum (compared to that of the best-level OFI), thus revealing its low rank structure.

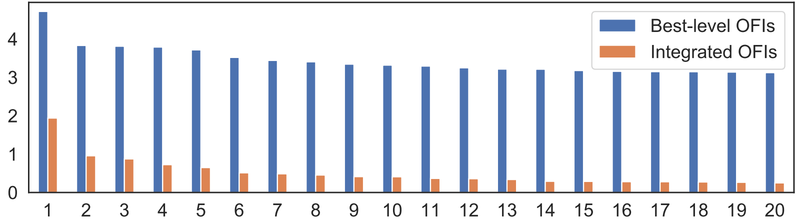
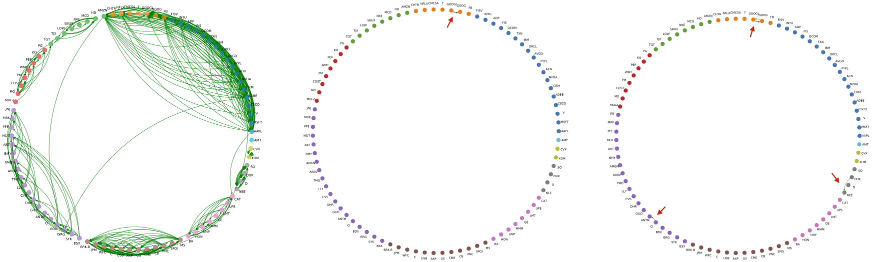


Figure 4: Barplot of singular values in descending order for the average coefficient matrix in $\mathbf{CI}^{[1]}$ and $\mathbf{CI}^{I}$. The coefficients are averaged over 2017–2019. We perform Singular Value Decomposition (SVD) on the coefficient matrix to obtain the singular values. The x-axis represents the singular value rank, and the y-axis represents the singular values.

The network of coefficients



(a) $\mathbf{CI}^{[1]}$, Threshold = 95-th percentile (b) $\mathbf{CI}^{I}$, Threshold = 95-th percentile (c) $\mathbf{CI}^{I}$, Threshold = 75-th percentile

Figure 5: Illustrations of the coefficient networks of $\mathbf{CI}^{[1]}$ and $\mathbf{CI}^{I}$. The coefficients are averaged over 2017–2019. To render the networks more interpretable and for ease of visualization, we only plot the top 5% largest (a-b), or top 25% largest (c) in magnitude coefficients. Thresholds are computed from **combined** coefficients of $\mathbf{CI}^{[1]}$ and $\mathbf{CI}^{I}$. Nodes are coloured by the GICS structure and sorted by market capitalization. The width of the edges is proportional to the absolute values of their respective coefficients.

Comparison with Capponi and Cont (2020)'s model

Capponi and Cont (2020) (CC) propose a two-step procedure to justify the significance of cross-impact terms

1. decompose $\mathrm{ofi}_{i,t}^{1,(h)}$ (the best-level OFIs) into a common factor $F_{\mathrm{ofi},t}^{(h)}$ and an idiosyncratic component $\tau_{i,t}^{(h)}$

$$\mathrm{ofi}_{i,t}^{1,(h)} = \mu_i + \gamma_i F_{\mathrm{ofi},t}^{(h)} + \tau_{i,t}^{(h)},$$

Comparison with Capponi and Cont (2020)'s model

Capponi and Cont (2020) (CC) propose a two-step procedure to justify the significance of cross-impact terms

1. decompose $\text{ofi}_{i,t}^{1,(h)}$ (the best-level OFIs) into a common factor $F_{\text{ofi},t}^{(h)}$ and an idiosyncratic component $\tau_{i,t}^{(h)}$

$$\text{ofi}_{i,t}^{1,(h)} = \mu_i + \gamma_i F_{\text{ofi},t}^{(h)} + \tau_{i,t}^{(h)},$$

2. regress individual returns on the common factor, its own idiosyncratic component and the idiosyncratic components of other stocks.

Comparison with Capponi and Cont (2020)'s model

Capponi and Cont (2020) (CC) propose a two-step procedure to justify the significance of cross-impact terms

1. decompose $\text{ofi}_{i,t}^{1,(h)}$ (the best-level OFIs) into a common factor $F_{\text{ofi},t}^{(h)}$ and an idiosyncratic component $\tau_{i,t}^{(h)}$

$$\text{ofi}_{i,t}^{1,(h)} = \mu_i + \gamma_i F_{\text{ofi},t}^{(h)} + \tau_{i,t}^{(h)},$$

2. regress individual returns on the common factor, its own idiosyncratic component and the idiosyncratic components of other stocks.

$$\begin{aligned}
\mathbf{PI}^{CC}: \; r_{i,t}^{(h)} &= \alpha_i^{CC} + \beta_{i0}^{CC} F_{\text{ofi},t}^{(h)} + \beta_{ii}^{CC} \tau_{i,t}^{(h)} + \epsilon_{i,t}^{CC}. \\
\mathbf{CI}^{CC}: \; r_{i,t}^{(h)} &= \alpha_i^{CC} + \beta_{i0}^{CC} F_{\text{ofi},t}^{(h)} + \beta_{ii}^{CC} \tau_{i,t}^{(h)} + \sum_{j \neq i} \beta_{ij}^{CC} \tau_{j,t}^{(h)} + \eta_{i,t}^{CC}.
\end{aligned} \tag{8}$$

Comparison with Capponi and Cont (2020)'s model

Capponi and Cont (2020) (CC) propose a two-step procedure to justify the significance of cross-impact terms

1. decompose $\text{ofi}_{i,t}^{1,(h)}$ (the best-level OFIs) into a common factor $F_{\text{ofi},t}^{(h)}$ and an idiosyncratic component $\tau_{i,t}^{(h)}$

$$\text{ofi}_{i,t}^{1,(h)} = \mu_i + \gamma_i F_{\text{ofi},t}^{(h)} + \tau_{i,t}^{(h)},$$

2. regress individual returns on the common factor, its own idiosyncratic component and the idiosyncratic components of other stocks.

$$\begin{aligned} \mathbf{PI}^{CC}: \ r_{i,t}^{(h)} &= \alpha_i^{CC} + \beta_{i0}^{CC} F_{\text{ofi},t}^{(h)} + \beta_{ii}^{CC} \tau_{i,t}^{(h)} + \epsilon_{i,t}^{CC}. \\ \mathbf{CI}^{CC}: \ r_{i,t}^{(h)} &= \alpha_i^{CC} + \beta_{i0}^{CC} F_{\text{ofi},t}^{(h)} + \beta_{ii}^{CC} \tau_{i,t}^{(h)} + \sum_{j \neq i} \beta_{ij}^{CC} \tau_{j,t}^{(h)} + \eta_{i,t}^{CC}. \end{aligned} \tag{8}$$

They demonstrate that as long as the common factor is involved in the model, adding cross-impact terms improves the explained proportion of the variance by only 0.5%.

	Best-level		Integrated		Best-level	
	$\mathbf{PI}^{[1]}$	$\mathbf{CI}^{[1]}$	$\mathbf{PI}^{I}$	$\mathbf{CI}^{I}$	$\mathbf{PI}^{CC}$	$\mathbf{CI}^{CC}$
In-Sample (%)	71.16	73.87	87.14	**87.85**	71.38	72.51
Out-Of-Sample (%)	64.64	66.03	**83.83**	83.62	63.01	63.60

Table 3: Comparison with Capponi and Cont (2020)'s model.

Our present study differs from CC in the following several aspects:

- (1) CC only focus on the in-sample performance. We consider both in-sample and out-of-sample performance, and place more emphasis on out-of-sample testing.

	Best-level		Integrated		Best-level	
	PI$^{[1]}$	**CI**$^{[1]}$	**PI**^{I}	**CI**^{I}	**PI**^{CC}	**CI**^{CC}
In-Sample (%)	71.16	73.87	87.14	**87.85**	71.38	72.51
Out-Of-Sample (%)	64.64	66.03	**83.83**	83.62	63.01	63.60

Table 3: Comparison with Capponi and Cont (2020)'s model.

Our present study differs from CC in the following several aspects:

- (1) CC only focus on the in-sample performance. We consider both in-sample and out-of-sample performance, and place more emphasis on out-of-sample testing.
- (2) CC ignore the fact that OLS becomes ill-posed when there are fewer observations than parameters, or potential multicollinearity issues in the cross-impact terms. Our model applies LASSO regularization to fit the regression and takes into account the potential sparsity of the cross-impact terms.

	Best-level		Integrated		Best-level	
	$\mathbf{PI}^{[1]}$	$\mathbf{CI}^{[1]}$	$\mathbf{PI}^{I}$	$\mathbf{CI}^{I}$	$\mathbf{PI}^{CC}$	$\mathbf{CI}^{CC}$
In-Sample (%)	71.16	73.87	87.14	**87.85**	71.38	72.51
Out-Of-Sample (%)	64.64	66.03	**83.83**	83.62	63.01	63.60

Table 3: Comparison with Capponi and Cont (2020)'s model.

Our present study differs from CC in the following several aspects:

- (1) CC only focus on the in-sample performance. We consider both in-sample and out-of-sample performance, and place more emphasis on out-of-sample testing.

- (2) CC ignore the fact that OLS becomes ill-posed when there are fewer observations than parameters, or potential multicollinearity issues in the cross-impact terms. Our model applies LASSO regularization to fit the regression and takes into account the potential sparsity of the cross-impact terms.

- (3) In addition to examining the cross-impact of best-level OFIs, we also consider the cross-impact from multi-level OFIs, in order to gauge a comprehensive understanding of the relations between multi-level OFIs of different assets and individual returns.

	Best-level		Integrated		Best-level	
	PI$^{[1]}$	**CI**$^{[1]}$	**PI**^{I}	**CI**^{I}	**PI**^{CC}	**CI**^{CC}
In-Sample (%)	71.16	73.87	87.14	**87.85**	71.38	72.51
Out-Of-Sample (%)	64.64	66.03	**83.83**	83.62	63.01	63.60

Table 3: Comparison with Capponi and Cont (2020)'s model.

Our present study differs from CC in the following several aspects:

- (1) CC only focus on the in-sample performance. We consider both in-sample and out-of-sample performance, and place more emphasis on out-of-sample testing.
- (2) CC ignore the fact that OLS becomes ill-posed when there are fewer observations than parameters, or potential multicollinearity issues in the cross-impact terms. Our model applies LASSO regularization to fit the regression and takes into account the potential sparsity of the cross-impact terms.
- (3) In addition to examining the cross-impact of best-level OFIs, we also consider the cross-impact from multi-level OFIs, in order to gauge a comprehensive understanding of the relations between multi-level OFIs of different assets and individual returns.
- (4) CC conclude that cross-impact is due to the common component in order flow across stocks. We claim that the main determinants of impact are from the idiosyncratic order flow imbalances at deeper levels.

Discussion about Cross-Impact

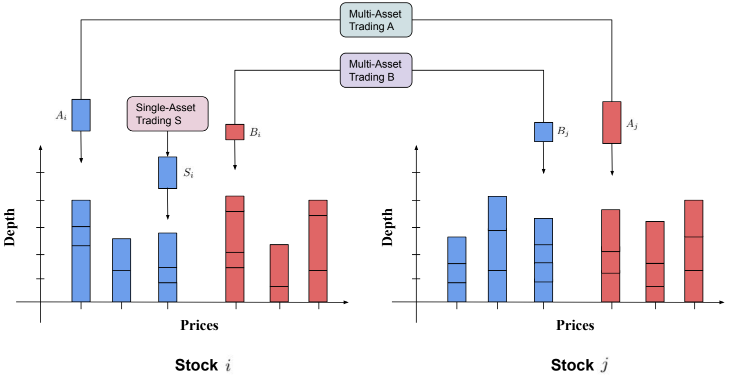
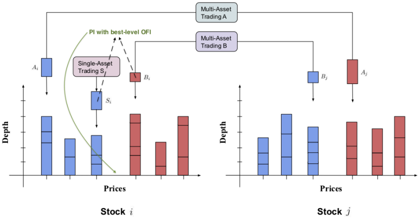
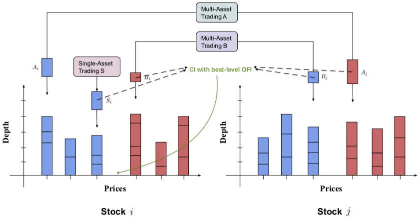


Figure 6: Illustration of the cross-impact model. The orders at different levels of each stock may come from single-asset and multi-asset trading strategies. The returns of stock i are potentially influenced by orders of stock j through the connections $A_j \to A_i \to \mathrm{ofi}_i^{\mathbf{3}} \to r_i$. Information along the path $A_j \to A_i \to \mathrm{ofi}_i^{\mathbf{3}} \to r_i$ can be collected by the Price-Impact model with integrated OFIs but not by the Price-Impact model with only best-level OFIs.

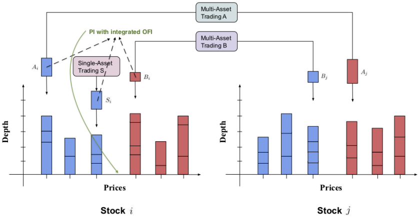
Discussion about Cross-Impact



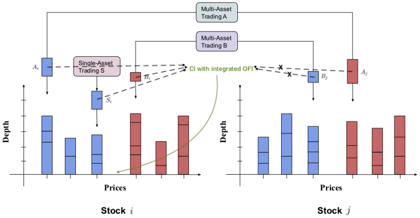
(a) **PI**[1]



(b) **CI**[1]



(c) **PI**^{I}



(d) **CI**^{I}

Outline

Forecasting Models

OFI-based models

$$\begin{aligned}
\textbf{FPI}^{[1]}: &\quad r_{i,t+1}^{(h)} = \alpha_i^{[1]} + \beta_i^{[1]} \mathrm{ofi}_{i,t}^{1,(h)} + \epsilon_{i,t+1}^{[1]}. \\
\textbf{FPI}^{I}: &\quad r_{i,t+1}^{(h)} = \alpha_i^{I} + \beta_i^{I} \mathrm{ofi}_{i,t}^{I} + \epsilon_{i,t+1}^{I}. \\
\textbf{FCI}^{[1]}: &\quad r_{i,t+1}^{(h)} = \alpha_i^{[1]} + \beta_{i,i}^{[1]} \mathrm{ofi}_{i,t}^{1,(h)} + \sum_{j \neq i} \beta_{i,j}^{[1]} \mathrm{ofi}_{j,t}^{1,(h)} + \eta_{i,t+1}^{[1]}. \\
\textbf{FCI}^{I}: &\quad r_{i,t+1}^{(h)} = \alpha_i^{I} + \beta_{i,i}^{I} \mathrm{ofi}_{i,t}^{I,(h)} + \sum_{j \neq i} \beta_{i,j}^{I} \mathrm{ofi}_{j,t}^{I,(h)} + \eta_{i,t+1}^{I}.
\end{aligned} \tag{9}$$

Return-based models

$$\begin{aligned}
\textbf{AR}: &\ r_{i,t+1}^{(h)} = \alpha_i + \beta_i r_{i,t}^{(h)} + \epsilon_{i,t+1}. \\
\textbf{CAR}: &\ r_{i,t+1}^{(h)} = \alpha_i + \beta_{i,i} r_{i,t}^{(h)} + \sum_{j \neq i} \beta_{i,j} r_{j,t}^{(h)} + \eta_{i,t+1}.
\end{aligned} \tag{10}$$

Chinco et al. (2019) considered the CAR model.

Empirical Results

Implementation

- In this experiment, observations associated with returns and OFIs are computed minutely, i.e. $h = 1$ minute.
- We use data from the previous 30 minutes to estimate the model parameters and apply the fitted model to forecast the one-minute-ahead return.
- We repeat this procedure for each trading day, to compute *rolling one-minute-ahead return forecasts* as in Chinco et al. (2019).

Empirical Results

Implementation

- In this experiment, observations associated with returns and OFIs are computed minutely, i.e. $h = 1$ minute.
- We use data from the previous 30 minutes to estimate the model parameters and apply the fitted model to forecast the one-minute-ahead return.
- We repeat this procedure for each trading day, to compute *rolling one-minute-ahead return forecasts* as in Chinco et al. (2019).

	Best-level		Integrated		Return	
	$\mathbf{FPI}^{[1]}$	$\mathbf{FCI}^{[1]}$	$\mathbf{FPI}^{I}$	$\mathbf{FCI}^{I}$	**AR**	**CAR**
In-Sample (%)	0.24	12.24	0.26	12.19	0.38	13.66
Out-Of-Sample (%)	-19.07	-11.98	-17.62	-11.81	-36.62	-11.19

Table 4: Statistical performance of forecasting models, including $\mathbf{FPI}^{[1]}$, $\mathbf{FPI}^{I}$, $\mathbf{FCI}^{[1]}$, $\mathbf{FCI}^{I}$, **AR** and **CAR**.

- Lagged cross-asset OFIs (returns) can help improve the forecast of future returns.
- Negative R^2 values do not imply that the forecasts are economically meaningless.

Strategy Profitability Analysis

To emphasize this aspect, we incorporate the return forecasts into two forecast-based trading strategies. We compute the strategy *profit and loss (PnL)* at each minute. We ignore trading costs, as this is not the focus of this paper.

- Forecast-implied portfolio (see Chinco et al. (2019))

$$w_{i,t} \stackrel{\text{def}}{=} \frac{1_{\{|f_{i,t}|>sprd_{i,t}\}} \cdot f_{i,t} \,/\, \sigma_{i,t}}{\sum_{n=1}^{N} 1_{\{|f_{n,t}|>sprd_{n,t}\}} \cdot |f_{n,t}| \,/\, \sigma_{n,t}}. \tag{11}$$

Strategy Profitability Analysis

To emphasize this aspect, we incorporate the return forecasts into two forecast-based trading strategies. We compute the strategy *profit and loss (PnL)* at each minute. We ignore trading costs, as this is not the focus of this paper.

- Forecast-implied portfolio (see Chinco et al. (2019))

$$w_{i,t} \overset{\text{def}}{=} \frac{1_{\{|f_{i,t}|>sprd_{i,t}\}} \cdot f_{i,t} \,/\, \sigma_{i,t}}{\sum_{n=1}^{N} 1_{\{|f_{n,t}|>sprd_{n,t}\}} \cdot |f_{n,t}| \,/\, \sigma_{n,t}}. \tag{11}$$

- Long-short portfolio

$$w_{i,t} \overset{\text{def}}{=} \frac{1_{\left\{f_{i,t}>d_t^{(\mathbf{9})}\right\}} - 1_{\left\{f_{i,t}<d_t^{(\mathbf{1})}\right\}}}{\sum_{n=1}^{N} \left[1_{\left\{f_{n,t}>d_t^{(\mathbf{9})}\right\}} + 1_{\left\{f_{n,t}<d_t^{(\mathbf{1})}\right\}}\right]}. \tag{12}$$

where $d_t^{(k)}$ denotes the k^{th} decile of the forecasted returns at time t.

Economic gains

PnL (bps)	Best-level FPI[1]	Best-level FCI[1]	Integrated FPI^{I}	Integrated FCI^{I}	Return AR	Return CAR
Forecast-implied	0.19	**0.99**	0.20	0.92	-0.14	0.88
Long-short	0.10	**0.91**	0.12	0.86	-0.29	0.58

Table 5: Economic gains of forecasting models, including **FPI**[1], **FPI**^{I}, **FCI**[1], **FCI**^{I}, **AR** and **CAR**.

- Portfolios based on forecasts of the predictive cross-impact model outperform those based on forecasts of the predictive price-impact model.

Economic gains

PnL (bps)	Best-level		Integrated		Return	
	FPI[1]	**FCI**[1]	**FPI**^{I}	**FCI**^{I}	**AR**	**CAR**
Forecast-implied	0.19	**0.99**	0.20	0.92	-0.14	0.88
Long-short	0.10	**0.91**	0.12	0.86	-0.29	0.58

Table 5: Economic gains of forecasting models, including **FPI**[1], **FPI**^{I}, **FCI**[1], **FCI**^{I}, **AR** and **CAR**.

- Portfolios based on forecasts of the predictive cross-impact model outperform those based on forecasts of the predictive price-impact model.
- Compared with models using the lagged returns as predictors, the models incorporating lagged OFIs yield higher PnLs, implying that OFIs might contain more predictive information than returns.

Outline

Conclusion

Our contributions can be summarized as follows:

1. We provide a systematic approach for combining OFIs at the top levels of the book into an integrated OFI variable which best explains price impact.

Conclusion

Our contributions can be summarized as follows:

1. We provide a systematic approach for combining OFIs at the top levels of the book into an integrated OFI variable which best explains price impact.
2. We show evidence that there is no need to introduce cross-impact terms for modeling the contemporaneous returns, as long as the OFI incorporates information from multiple levels, not just the best level.

Conclusion

Our contributions can be summarized as follows:

1. We provide a systematic approach for combining OFIs at the top levels of the book into an integrated OFI variable which best explains price impact.
2. We show evidence that there is no need to introduce cross-impact terms for modeling the contemporaneous returns, as long as the OFI incorporates information from multiple levels, not just the best level.
3. Cross-asset OFIs improve the forecast of future returns, thus providing evidence of cross-impact in the forecasting model.

References I

Bouchaud, J.-P. (2010). Price impact. *Encyclopedia of Quantitative Finance*.

Capponi, F. and Cont, R. (2020). Multi-asset market impact and order flow commonality. *Available at SSRN*.

Chinco, A., Clark-Joseph, A. D., and Ye, M. (2019). Sparse signals in the cross-section of returns. *Journal of Finance*, 74(1):449–492.

Cont, R., Kukanov, A., and Stoikov, S. (2014). The price impact of order book events. *Journal of Financial Econometrics*, 12(1):47–88.

Eisler, Z., Bouchaud, J.-P., and Kockelkoren, J. (2012). The price impact of order book events: market orders, limit orders and cancellations. *Quantitative Finance*, 12(9):1395–1419.

Xu, K., Gould, M. D., and Howison, S. D. (2018). Multi-level order-flow imbalance in a limit order book. *Market Microstructure and Liquidity*, 4(03n04):1950011.