GRAPH DATA SCIENCE: THE TWO CULTURES

Deep Mukhopadhyay and Kaijun Wang

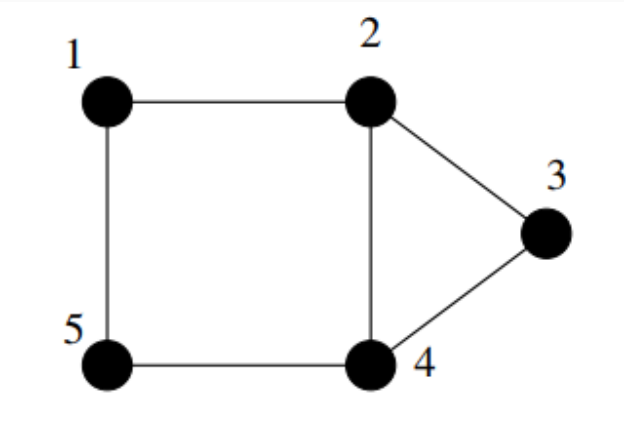
Data Science Seminar

March 12, 2019

What is Graph?

An undirected graph $\mathcal{G} = (V, E)$ consists of:

- Nodes/Vertices $V = \{1, \cdots, n\}$
- Edges $E = \{(i, j), i, j \in V\}$
- Adjacency matrix A, where A_{ij} represent edges, A is non-negative and symmetric

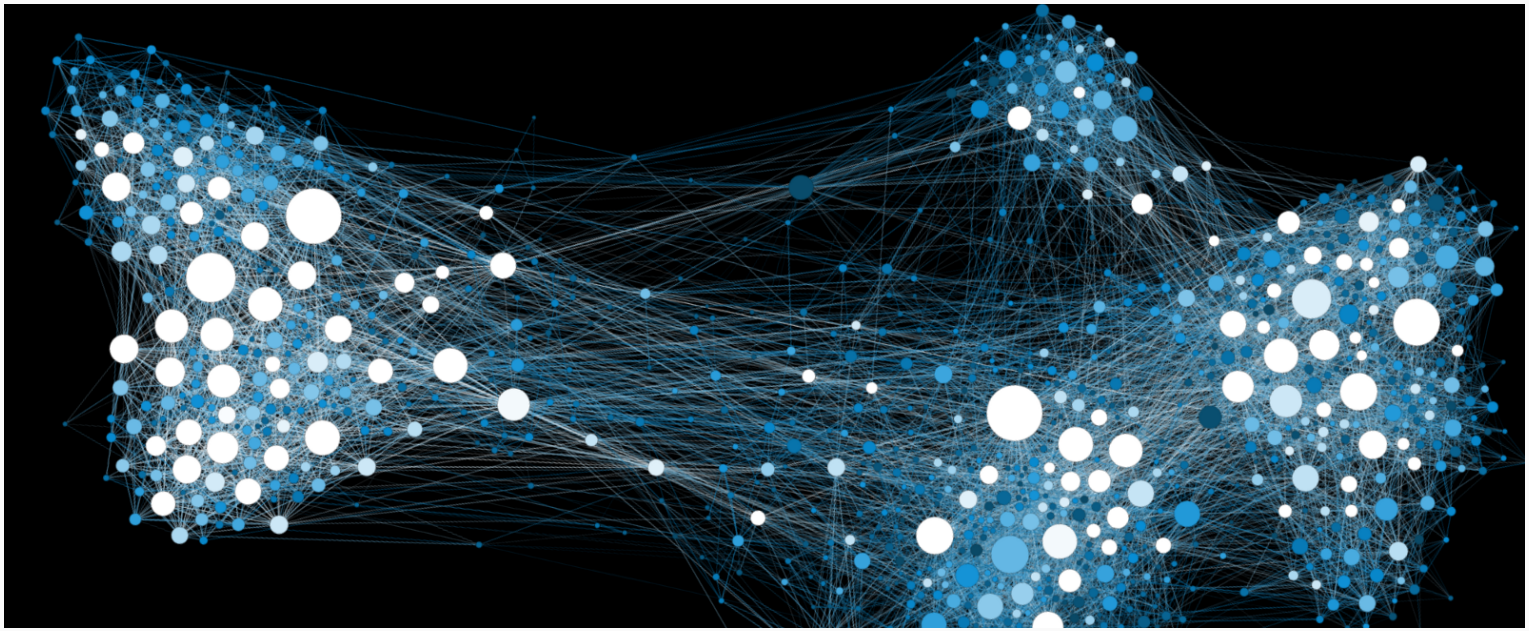


$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

- Vertex degree $d_j = \sum_i A_{ij},\ j = 1, \cdots, n$; degree matrix $D \in \mathbb{R}^{n \times n}$ is the diagonal matrix with elements $(d_1, ..., d_n)$.

Why Graphs?

- Graph-data: Complex networks are everywhere in sciences and engineering: brain networks, protein-protein interaction network, spatial networks, social networks, and the World Wide Web etc.



Graph Data Science: The 'Big Tent'

- Graph-data: Complex networks are everywhere in sciences and engineering: biological networks, brain networks, transportation networks, social networks, and the World Wide Web etc.

- Data-graph: Graphs serve as an abstract model for various types of data structures including high-dimensional or even object data like image /text; Towards an elegant solution to the big data variety problem. The basic idea consists of two steps:

Graph Data Science: The 'Big Tent'

- Graph-data: Complex networks are everywhere in sciences and engineering: biological networks, brain networks, transportation networks, social networks, and the World Wide Web etc.

- Data-graph: Graphs serve as an abstract model for various types of data structures including high-dimensional or even object data like image /text; Towards an elegant solution to the big data variety problem. The basic idea consists of two steps:

`Step 1`. [representation step] code the given data using graphs.

`Step 2`. [analysis step] reformulate and apply statistical methods on the graph-transformed problem, which often is much easier than the original problem.

US Senate Voting Data: Graph-based Nonparametrics

- Data: Voting records of the US Senate covering the period from George H. W. Bush term (Jan 1989) to the end of Bill Clinton's term (Jan 2001).
- $n = 2678$ bills were submitted over that period, binary voting decisions (1 for yes, 0 for no) were recorded from each of the 100 seats.

Time	Bills	Senate Seats						
		1	2	3	$\cdots$	98	99	100
1989-02-28	1	0	1	1	$\cdots$	1	0	1
1989-02-28	2	0	1	0	$\cdots$	1	0	0
$\vdots$		$\vdots$		$\vdots$		$\vdots$		$\vdots$
2000-12-05	2677	1	1	1	$\cdots$	0	1	1
2000-12-07	2678	1	1	1	$\cdots$	0	1	1

Goal: Find the change-point for the voting patterns. HD Nonparametric change-point detection is known to be a notoriously difficult problem.

US Senate Voting Data: Graph-based Nonparametrics

- Data: Voting records of the US Senate covering the period from George H. W. Bush term (Jan 1989) to the end of Bill Clinton's term (Jan 2001).
- $n = 2678$ bills were submitted over that period, binary voting decisions (1 for yes, 0 for no) were recorded from each of the 100 seats.

Time	Bills	Senate Seats						
		1	2	3	$\cdots$	98	99	100
1989-02-28	1	0	1	1	$\cdots$	1	0	1
1989-02-28	2	0	1	0	$\cdots$	1	0	0
$\vdots$		$\vdots$		$\vdots$		$\vdots$		$\vdots$
2000-12-05	2677	1	1	1	$\cdots$	0	1	1
2000-12-07	2678	1	1	1	$\cdots$	0	1	1

Goal: Find the change-point for the voting patterns. HD Nonparametric change-point detection is known to be a notoriously difficult problem.

The TRICK is to: Convert this into a graph problem.

Data Domain to Graph Domain

1. Vertices: Construct the network with vertices V as the **time points** (or, equivalently, the bills).
2. Edges: Compute **Pearson-ϕ^2** coeffs to measure the level of agreement between two voting records.
3. A: Construct the **weighted** adjacency matrix A.
4. Reformulate: Change point detection as the problem of finding disjoint communities in a network.

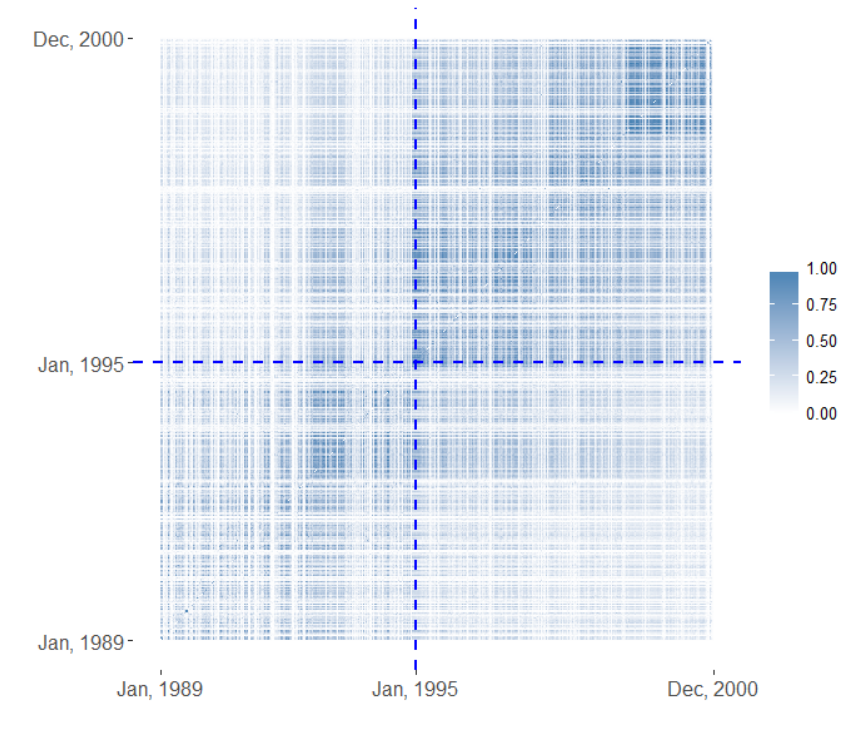


Figure: The weighted adjacency matrix A. Dashed blue lines indicates the change point at January, 1995.

PRACTICE OF SPECTRAL GRAPH ANALYSIS

Algorithms: How To Analyze Graphs?

Spectral Graph Theory: It provides an elegant framework and a formal mathematical language for understanding patterns of interconnectedness in a graph, which has produced impressive results across a range of application domains.

1. Find a "suitable" spectral graph matrix. Common choices are:
   - $\mathcal{L} = D^{-1/2} A D^{-1/2}$ Chung (1997)
   - $\mathcal{B} = A - N^{-1} \boldsymbol{d}\boldsymbol{d}^T$ Newman (2006)
   - $\mathcal{T} = D^{-1} A$ Coifman and Lafon (2006)
   - Type-I Reg. $\mathcal{L}_\tau = D_\tau^{-1/2} A\, D_\tau^{-1/2}$ Chaudhuri et al. (2012)
   - Type-II Reg. $\mathcal{L}_\tau = D_\tau^{-1/2} A_\tau\, D_\tau^{-1/2}$ Amini et al. (2013)
   - Google's PageRank $\mathcal{T}_\alpha = \alpha D^{-1} A + \frac{1}{n}(1-\alpha)\mathbf{1}\mathbf{1}^T$ Brin and Page (1999)
2. Graph Fourier Basis: Compute singular vectors of step 1 selected matrix.
3. Graph signal processing: Perform learning tasks such as regression, clustering, smoothing, kriging by expanding data (defined over the vertices of a graph) on the selected graph Fourier basis.

K-Communities in the Senate Network

Algorithm:

1. Obtain the Laplacian matrix $\mathcal{L} = D^{-1/2}AD^{-1/2}$ from A.
2. Perform SVD on $\mathcal{L}$, extract top $K-1$ dominant singular vectors $U = [u_2, \cdots, u_k]$.

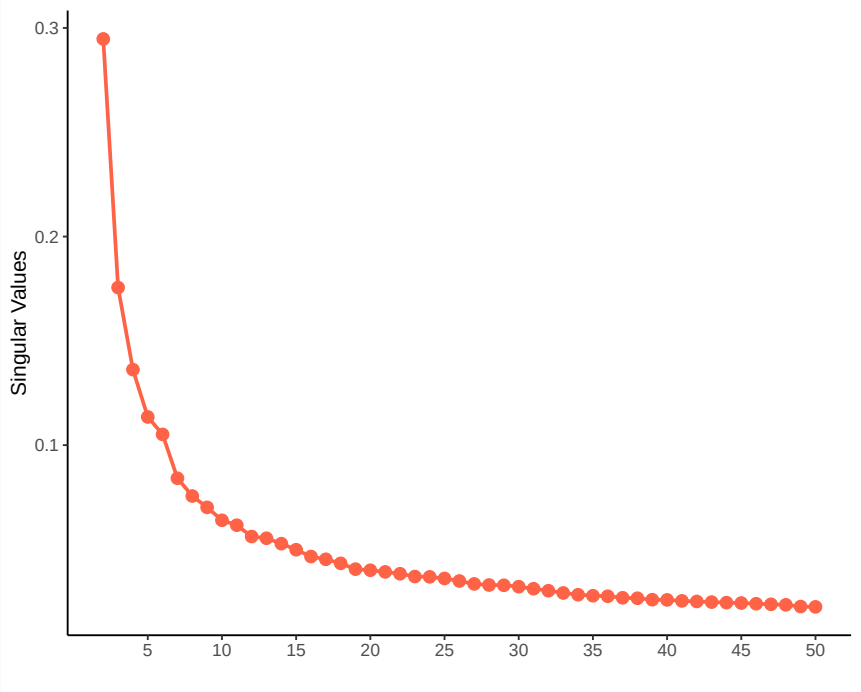


Figure: Top Laplacian singular values of Senate Data, there's a clear gap between second and subsequent singular values.

K-Communities in the Senate Network

Algorithm:

1. Obtain the Laplacian matrix $\mathcal{L} = D^{-1/2}AD^{-1/2}$ from A.
2. Perform SVD on $\mathcal{L}$, extract top $K-1$ dominant singular vectors $U = [u_2, \cdots, u_k]$.
3. Perform k-means clustering on the rows of U to obtain K groups.

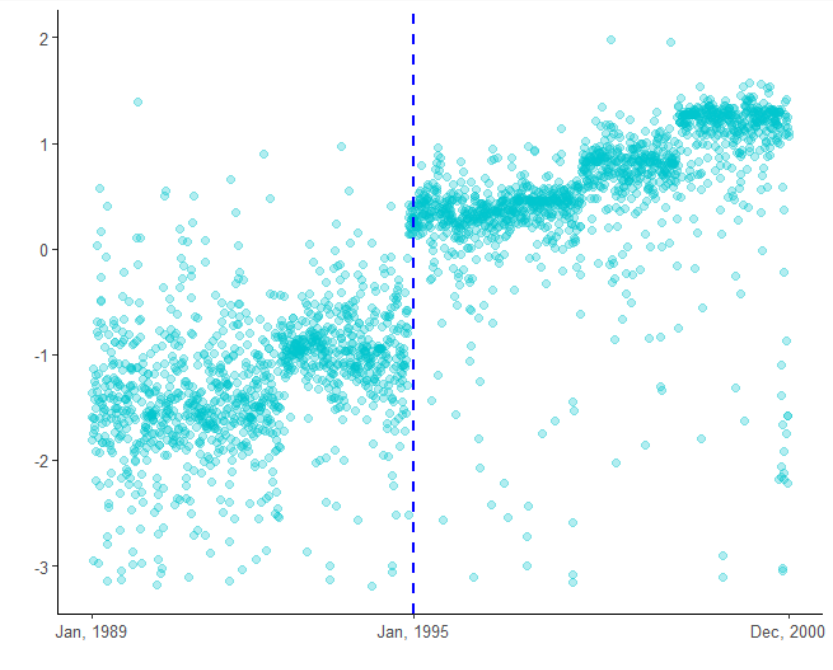


Figure: Top Laplacian singular vector u_2 of Senate Data, shown in blue dots.

THE STATISTICAL AND COMPUTATIONAL CHALLENGES

Two Challenges

Classical spectral methods almost immediately hit a wall. The challenge comes from two principal directions:

- Statistical: The spiky Laplacian embedding completely fails to segment the time periods into homogeneous blocks to identify the 'smooth' transition of the voting patterns around Jan, 1995.

- Computational: Existing spectral graph algorithms are awfully expensive, with an $O(n^2)$ cost. In particular, explicitly computing the SVD of the $n \times n$ Laplacian matrix may not be practical or even feasible for massive-scale problems.

Missing the Forest for the Trees

The way in which spectral graph analysis is currently taught and practiced is:

- rather mechanical: consisting of a series of matrix calculations.
- very much unsystematic: The diversity of poorly related graph learning algorithms attests to this fact; Its like a bag of tricks.

This kind of dry mechanical treatment a huge negative bearing on our understanding; provides absolutely no clue (other than making blind guesses) on how to adapt the existing theory and algorithms for yet-unseen complex graph problems.

Half-a-century-old Problem of Network Science

A brand new perspective is needed, not just modification of the old. Where “Beauty” and “Utility” Can Coexist.

Half-a-century-old Problem of Network Science

A brand new perspective is needed, not just modification of the old. Where “Beauty” and “Utility” Can Coexist.

1. Beauty: That it confers the power to **unify** disparate methods of graph data analysis using one *single* formalism and algorithm.

Half-a-century-old Problem of Network Science

A brand new perspective is needed, not just modification of the old. Where “Beauty” and “Utility” Can Coexist.

1. Beauty: That it confers the power to **unify** disparate methods of graph data analysis using one *single* formalism and algorithm.

2. Utility: That it opens up possibilities for **constructing specially designed** more efficient practical algorithms for non-standard networks, such as the Senate data example.

TOWARDS A NEW STATISTICAL FOUNDATION

Probability Notation and Definitions

1. Network Probability Mass Function:

$$P(x, y; \mathcal{G}) = A(x, y; \mathcal{G}) / \sum_{x,y} A(x, y; \mathcal{G}).$$

2. Vertex Probability Mass Function:

$$p(x; \mathcal{G}) = \sum_{y} P(x, y; \mathcal{G}), \quad \text{and} \quad p(y; \mathcal{G}) = \sum_{x} P(x, y; \mathcal{G}).$$

3. Graph Interaction Function (GIF) is the ratio of

$$\mathrm{GIF}(x, y; \mathcal{G}) = \frac{P(x, y; \mathcal{G})}{p(x; \mathcal{G}) p(y; \mathcal{G})}, \quad x, y \in V(\mathcal{G})$$

4. Graph Correlation Density Field (GraField) is a piecewise-constant bivariate kernel function $\mathscr{C} : [0, 1]^2 \to \mathbb{R}_+ \cup \{0\}$

$$\mathscr{C}(u, v; \mathcal{G}) = \mathrm{GIF}\big[Q(u; X), Q(v; Y); \mathcal{G}\big] = \frac{p\big(Q(u; X), Q(v; Y); \mathcal{G}\big)}{p\big(Q(u; X)\big) p\big(Q(v; Y)\big)}, \quad 0 < u, v < 1.$$

where $u = F(x; \mathcal{G})$, $v = F(y; \mathcal{G})$, and $Q(u; X)$ and $Q(u; Y)$ are the respective quantile functions.

Karhunen-Loéve Representation of Graph

We define the Karhunen-Loéve (KL) representation of a graph $\mathcal{G}$ based on the spectral expansion of its `GraField` function $\mathscr{C}(u, v; \mathcal{G})$:

$$\mathscr{C}(u, v; \mathcal{G}) = 1 + \sum_{k=1}^{n-1} \lambda_k \phi_k(u) \phi_k(v), \tag{1}$$

where the non-negative $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_{n-1} \geq 0$ are singular values and $\{\phi_k\}_{k \geq 1}$ are the orthonormal singular functions $\langle \phi_j, \phi_k \rangle_{\mathscr{L}^2[0,1]} = \delta_{jk}$, for $j, k = 1, \ldots, n-1$, satisfying the following integral equation equation:

$$\int_{[0,1]} [\mathscr{C}(u, v; \mathcal{G}) - 1] \, \phi_k(v) \, \mathrm{d}v = \lambda_k \phi_k(u), \quad k = 1, 2, \ldots, n-1. \tag{2}$$

A Nonparametric Statistical View

• A Statistical Reformulation: Spectral graph theory can now be viewed as the following statistical estimation problem where the goal is to approximate $(\lambda_k, \phi_k)_{k\geq 1}$ that satisfy the integral equation (2)

$$A_{n\times n} \;\mapsto\; \mathscr{C} \;\mapsto\; \left\{(\widehat{\lambda}_1, \widehat{\phi}_1), \ldots, (\widehat{\lambda}_{n-1}, \widehat{\phi}_{n-1})\right\} \text{ that satisfies Eq. (2).}$$

• Projection Methods for Eigenvector Approximation: We approximate the unknown eigenvectors by the projecting ϕ_k on the $\mathsf{span}\{\eta_j, j = 1, \ldots, n\}$ defined by

$$\phi_k(u) \approx \mathcal{P}_n \phi_k = \sum_{j=1}^{n} \theta_{jk}\, \eta_j(u), \quad 0 < u < 1 \tag{3}$$

where θ_{jk} are the unknown coefficients <u>to be estimated</u>.

Theory of Approximation: A Rough Sketch

`Step 1.` The residual of the governing integral equation (2)

$$R_k(u) \equiv \sum_j \theta_{jk}\Big[\int_0^1 (\mathscr{C}(u,v;\mathcal{G})-1)\eta_j(v)\,\mathrm{d}v - \lambda_k\eta_j(u)\Big] = 0.$$

`Step 2.` Requiring $R_k(u)$ is orthogonal to each of the basis functions $\langle R_k(u), \eta_l(u)\rangle_{\mathscr{L}^2[0,1]} = 0$ for $k = 1, \ldots, n$ lead to

$$\sum_j \theta_{jk}\Big[\iint_{[0,1]^2} (\mathscr{C}(u,v;\mathcal{G})-1)\eta_j(v)\eta_l(u)\,\mathrm{d}v\,\mathrm{d}u\Big] - \lambda_l \sum_j \theta_{jl}\Big[\int_0^1 \eta_j(u)\eta_l(u)\,\mathrm{d}u\Big] = 0.$$

`Step 3.` Represent these system compactly in the matrix format:

The Master Equation : $\mathcal{M}\Theta = H\Theta\Delta.$

The G-Matrix

• A generalized spectral matrix associated with the graph: G-matrix. Define discrete graph transform with respect to an orthonormal system η as

$$\mathcal{M}[j,k;\eta,\mathcal{G}] \;=\; \left\langle \eta_j, \int_0^1 (\mathscr{C}-1)\eta_k \right\rangle_{\mathscr{L}^2[0,1]} \quad \text{for } j,k=1,\ldots,n. \qquad (4)$$

• This can also be interpreted as the transform coefficient matrix of the orthogonal series expansion of the GraField kernel $\mathscr{C}(u,v;\mathcal{G})$ with respect to the product bases $\{\eta_j \otimes \eta_k\}_{1\le j,k\le n}$.

• It provides a systematic recipe for converting the graph problem into a "suitable" matrix problem:

$$\mathcal{G}(V,E) \;\longrightarrow\; A_{n\times n} \;\longrightarrow\; \mathscr{C}(u,v;\mathcal{G}) \;\xrightarrow[\text{Eq. (4)}]{\{\eta_1,\ldots,\eta_n\}}\; \mathcal{M}(\eta,\mathcal{G}) \in \mathbb{R}^{n\times n}.$$

The Key Result

Theorem (Nonparametric spectral approximation)

The Fourier coefficients $\{\theta_{jk}\}$ of the projection estimators (3) of the GraField eigenfunctions (eigenvalues and eigenvectors), satisfying the integral equation (2), can be obtained by solving the following generalized matrix eigenvalue problem:

$$\mathcal{M}\Theta = H\Theta\Delta, \tag{5}$$

where $\mathcal{M}_{jk} = \left\langle \eta_j, \int_0^1 (\mathscr{C}_n - 1)\eta_k \right\rangle_{\mathscr{L}^2[0,1]}$, $\Theta_{jk} = \theta_{jk}$, $\Delta_{jk} = \delta_{jk}\lambda_k$, *and* $H_{jk} = \left\langle \eta_j, \eta_k \right\rangle_{\mathscr{L}^2[0,1]}$.

The Key Result

Theorem (Nonparametric spectral approximation)

The Fourier coefficients $\{\theta_{jk}\}$ of the projection estimators (3) of the GraField eigenfunctions (eigenvalues and eigenvectors), satisfying the integral equation (2), can be obtained by solving the following generalized matrix eigenvalue problem:

$$\mathcal{M}\Theta = H\Theta\Delta, \tag{5}$$

where $\mathcal{M}_{jk} = \left\langle \eta_j, \int_0^1 (\mathscr{C}_n - 1)\eta_k \right\rangle_{\mathscr{L}^2[0,1]}$, $\Theta_{jk} = \theta_{jk}$, $\Delta_{jk} = \delta_{jk}\lambda_k$, *and* $H_{jk} = \left\langle \eta_j, \eta_k \right\rangle_{\mathscr{L}^2[0,1]}$.

Let's convert this general theory into a concrete algorithm.

Unified Statistical Algorithm

`Step 1.` Choose $\{\xi_j\}$ to be an orthonormal basis of $\mathscr{L}^2(F)$ Hilbert space (The vertex probability measure) satisfying

$$\sum_x \xi_j(x; F, \mathcal{G})\, \xi_k(x; F, \mathcal{G})\, p(x; \mathcal{G}) \;=\; 0 \;\text{ for } j \neq k.$$

`Step 2.` $\xi \mapsto \eta$ via Quantile Transform: Construct $\eta_j(u) := \xi_j(Q(u; X))$ on the unit interval $0 \leq u \leq 1$.

`Step 3.` Transform coding of graphs. Construct generalized spectral graph matrix or `G-matrix` $\mathcal{M}(\xi; \mathcal{G}) \in \mathbb{R}^{n \times n}$:

$$\mathcal{M}[j, k; \xi, \mathcal{G}] \;=\; \left\langle \eta_j, \int_0^1 (\mathscr{C} - 1)\eta_k \right\rangle_{L^2[0,1]} = \sum_{\ell, m} \xi_j(\ell; F)\xi_k(m; F)P(\ell, m; \mathcal{G}) \quad (6)$$

$\mathcal{M}(\xi; \mathcal{G})$ can be viewed as a *orthogonal transform coefficient matrix* of $\mathcal{G}$ w.r.t the product bases $\{\eta_j \otimes \eta_k\}_{1 \leq j,k \leq n}$.

`Step 4.` Perform the SVD of $\mathcal{M}(\xi;\mathcal{G}) = U\Gamma U^T = \sum_k u_k \gamma_k u_k^T$, where u_{ij} are the elements of the singular vector matrix $U = (u_1, \ldots, u_{n-1})$, and $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_{n-1})$, $\gamma_1 \geq \cdots \gamma_{n-1} \geq 0$.

`Step 5.` Approximated singular values $\widetilde{\lambda}_k = \gamma_k$ for $k = 1, \ldots, n-1$.

`Step 6.` Obtain Karhunen-Loéve (KL) graph representation bases $\widetilde{\phi}_k$ by taking the following linear combination:

$$\widetilde{\phi}_k = \sum_{j=1}^{n} u_{jk}\xi_j, \ \text{ for } k = 1, \ldots, n-1,$$

which can be directly used for subsequent data analysis on graphs.

`Step 4.` Perform the SVD of $\mathcal{M}(\xi;\mathcal{G}) = U\Gamma U^T = \sum_k u_k \gamma_k u_k^T$, where u_{ij} are the elements of the singular vector matrix $U = (u_1, \ldots, u_{n-1})$, and $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_{n-1})$, $\gamma_1 \geq \cdots \gamma_{n-1} \geq 0$.

`Step 5.` Approximated singular values $\widetilde{\lambda}_k = \gamma_k$ for $k = 1, \ldots, n-1$.

`Step 6.` Obtain Karhunen-Loéve (KL) graph representation bases $\widetilde{\phi}_k$ by taking the following linear combination:

$$\widetilde{\phi}_k = \sum_{j=1}^{n} u_{jk}\xi_j, \ \text{ for } k = 1, \ldots, n-1,$$

which can be directly used for subsequent data analysis on graphs.

We will see that **All** known spectral graph techniques are just different manifestations of this single general algorithm.

NONPARAMETRIC SYNTHESIS

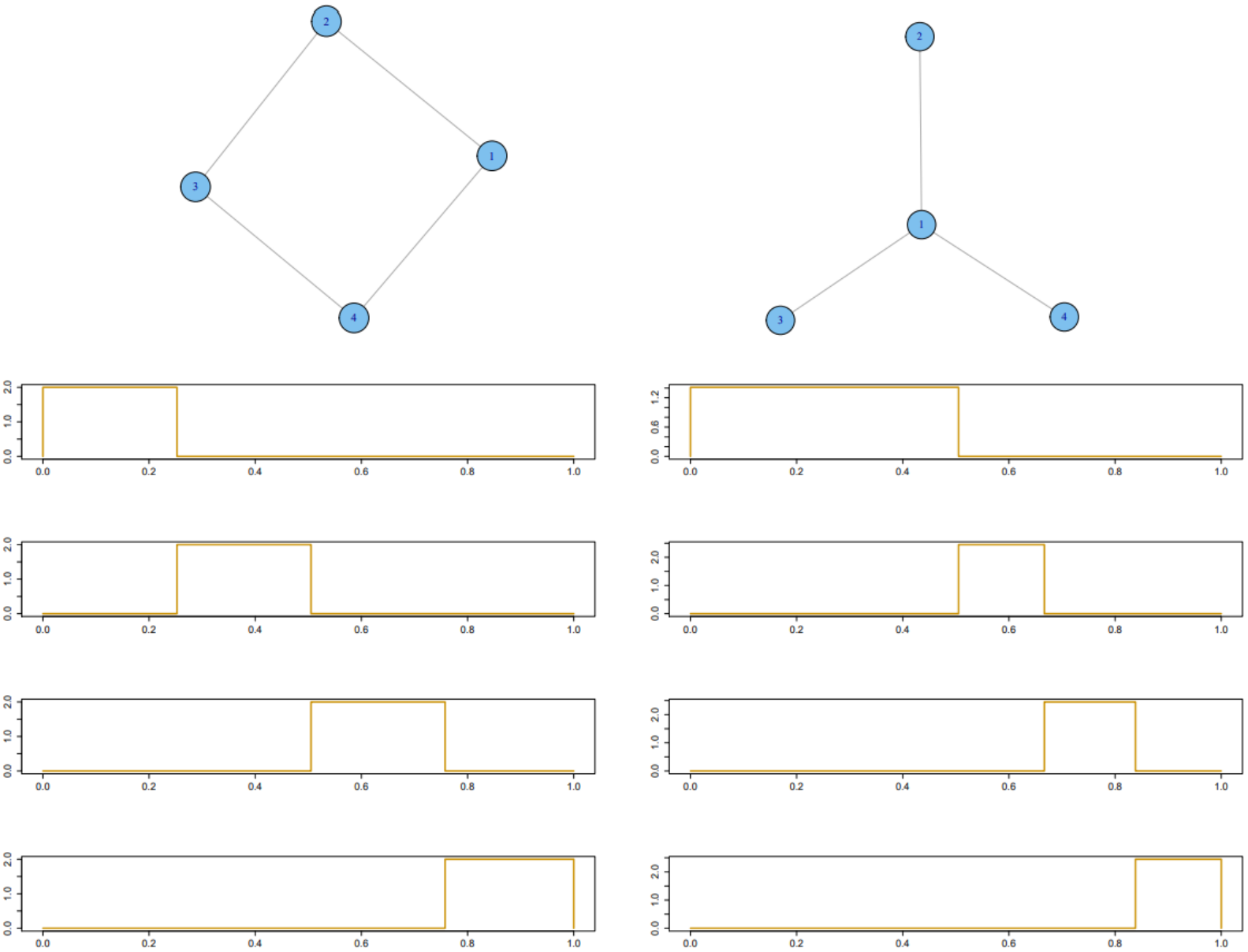
Example 1: Laplacian Spectral Analysis

Choose η to be **Degree-Adaptive Block-pulse functions** defined on the non-uniform grid $0 = u_0 < u_1 \cdots < u_n = 1$ over [0,1], where $u_j = F(j; \mathcal{G})$ with local support

$$\eta_j(u) = \begin{cases} p^{-1/2}(j) & \text{for } u_{j-1} < u \leq u_j; \\ 0 & \text{elsewhere.} \end{cases} \tag{7}$$

They are disjoint, orthogonal, and a complete set of functions satisfying

$$\int_0^1 \eta_j(u)\,\mathrm{d}u = \sqrt{p(j)}, \quad \int_0^1 \eta_j^2(u)\,\mathrm{d}u = 1, \quad \text{and} \quad \int_0^1 \eta_j(u)\eta_k(u)\,\mathrm{d}u = \delta_{jk}.$$


2
1
3
4
2
1
3
4
0.0
1.0
2.0
0.0
0.6
1.2
0.0
0.2
0.4
0.6
0.8
1.0

What Spectral Method We Get Out Of It?

Choose η to be **Degree-Adaptive Block-pulse functions** defined on the non-uniform grid $0 = u_0 < u_1 \cdots < u_n = 1$ over [0,1], where $u_j = F(j; \mathcal{G})$ with local support

$$\eta_j(u) = \begin{cases} p^{-1/2}(j) & \text{for } u_{j-1} < u \leq u_j; \\ 0 & \text{elsewhere.} \end{cases} \tag{8}$$

They are disjoint, orthogonal, and a complete set of functions satisfying

$$\int_0^1 \eta_j(u)\,\mathrm{d}u = \sqrt{p(j)}, \quad \int_0^1 \eta_j^2(u)\,\mathrm{d}u = 1, \quad \text{and} \quad \int_0^1 \eta_j(u)\eta_k(u)\,\mathrm{d}u = \delta_{jk}.$$

Let The Master Equation Decide : $\mathcal{M}\Theta = H\Theta\Delta$.

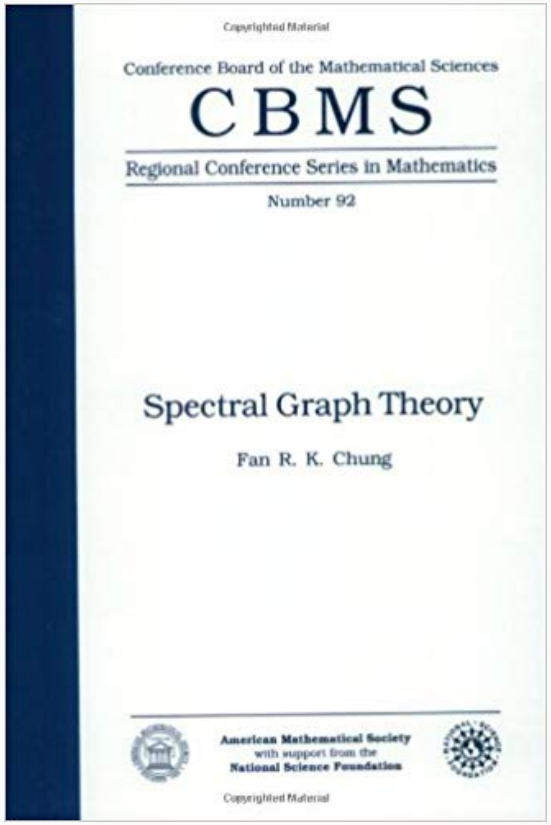
Statistical Derivation of Laplacian Method

Theorem (Laplacian Spectral Analysis)

Let $\phi_1, \ldots, \phi_n$ the canonical Schmidt bases of $\mathscr{L}^2$ graph kernel $\mathscr{C}(u, v; \mathcal{G})$, satisfying the integral equation (2). Then the empirical solution of (2) for Fourier coefficients $\{\theta_{jk}\}$ approximated by block-pulse orthogonal series (8) can equivalently be written down in closed form as the following matrix eigen-value problem

$$\boxed{\textit{Normalized Laplacian}: \quad \mathcal{L}_0 \Theta = \Theta \Lambda,} \tag{9}$$

where $\mathcal{L}_0 = \mathcal{L} - uu^T$, $\mathcal{L}$ is the Laplacian matrix, $u = D_p^{1/2} 1_n$, and $D_p = \mathrm{diag}(p_1, \ldots, p_n)$.



The normalized-Laplacian matrix was introduced by *mathematician* Fan Chung (1992) using combinatorial arguments.

Example 2: Modularity Spectral Analysis

Theorem (Modularity Spectral Analysis)

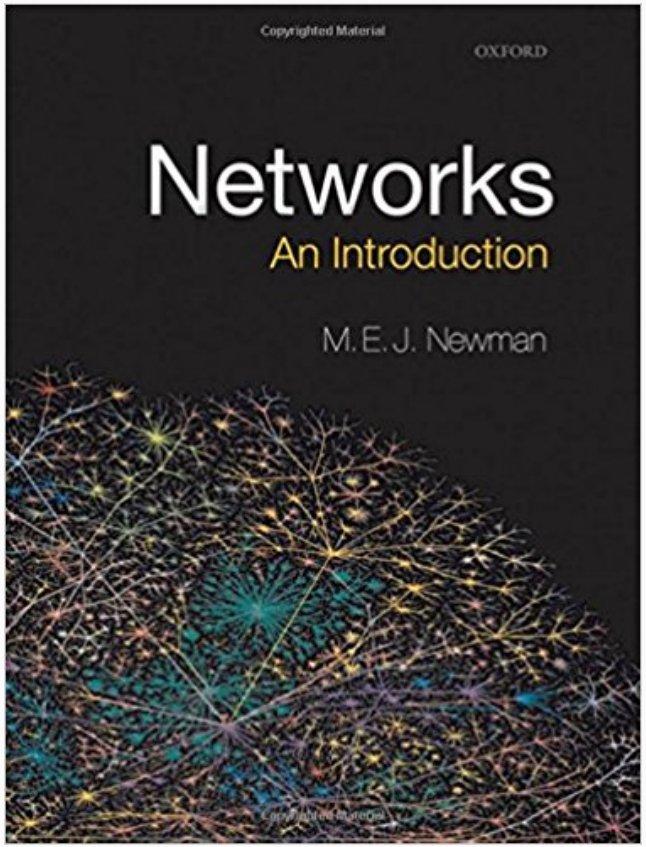
To approximate the Karhunen-Loéve graph basis $\phi_k = \sum_j \theta_{jk}\eta_j$, choose $\eta_j(u) = \mathbf{I}(u_{j-1} < u \leq u_j)$ to be the characteristic function satisfying

$$\int_0^1 \eta_j(u)\,\mathrm{d}u \;=\; \int_0^1 \eta_j^2(u)\,\mathrm{d}u \;=\; p(j;\mathcal{G}_n).$$

Then our "Master Equation" reduces to the following generalized eigenvalue equation:

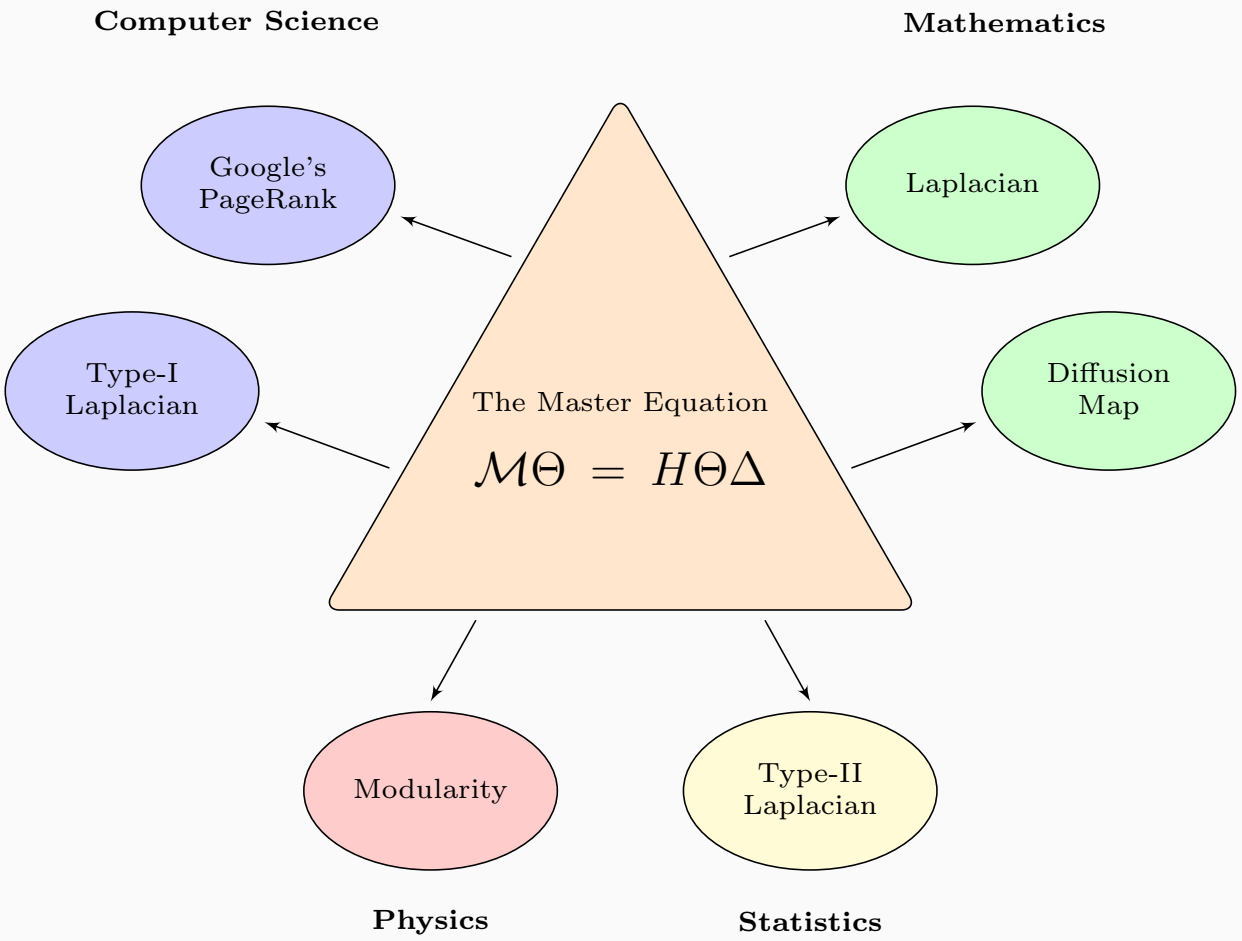
$$\boxed{\text{Modularity matrix}: \qquad \mathcal{B}\Theta \;=\; D\Theta\Lambda,} \tag{10}$$

where the matrix $\mathcal{B}$ is given by $A - \frac{1}{\mathrm{Vol}(\mathcal{G})}dd^T$, $\mathrm{Vol}(\mathcal{G}) = \sum_{x,y} A(x,y;\mathcal{G})$, and $d = A1_n$ is the degree vector.



The modularity matrix $\mathcal{B}$ was introduced by the *physicist* Newman (2006) from an entirely different motivation.

Logically connected means to reach different ends



Traditional spectral graph techniques **"naturally emerge"** from the Master Equation in a self-consistent manner.

The Age of 'Unified Algorithms' Is Here: The Bigger Picture

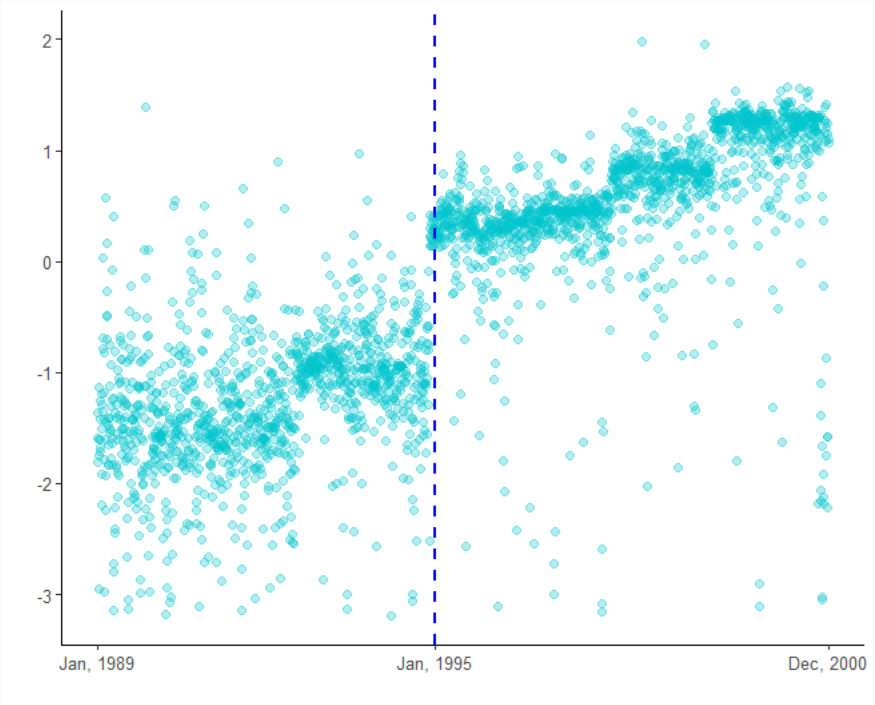
What Is the Broader Theme? Despite a wealth of algorithms, we still have relatively few, if any, global theories of data analysis.

> *"Assuming that a unified foundation is inevitable, what will it be? I think the general refusal in our field to strive for a unified perspective has been the single biggest impediment to its advancement"* – *Jim Berger*.

Paradigm of "United Statistical Algorithms": The goal is to develop a universal language (a general principle) of data analysis that can reveal the interconnectedness among different branches of statistics – *one of the fundamental open questions of 21st century statistics.*

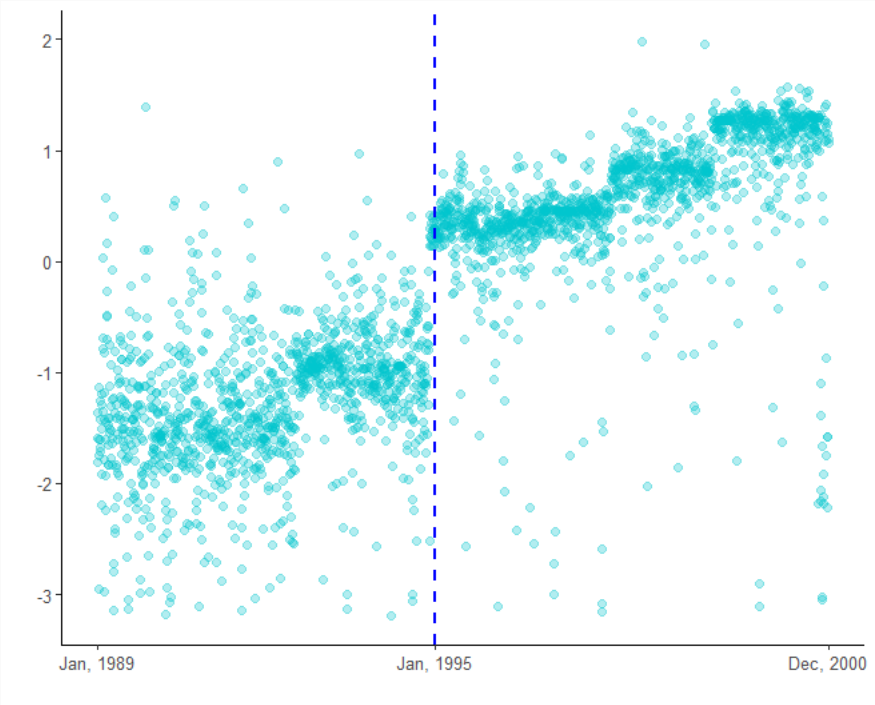
GOING BEYOND CLASSICAL METHODS

Revisiting Senate Data Example



- Challenge: Better (*faster and more accurate*) approximation of the Laplacian embedding shown in blue dots.

Revisiting Senate Data Example



- Challenge: Better (*faster and more accurate*) approximation of the Laplacian embedding shown in blue dots. HOW?

- Can our new perspective help us solving this riddle?

Intuition

- First note that the Senate-graph is endowed with a special structure, not an arbitrary one.

- The nodes have some kind of natural ordering (time-points), which induces a smoothly varying shape in the Laplacian singular vector.

- An obvious goal: How to intelligently design a trial-basis function

$$\widehat{\phi}_k \approx \sum_{j=1}^{m} \theta_{jk} S_j(u), \quad k = 1, \ldots, k_0$$

(i) Compressibility: Need fewer terms $m \ll n$ to achieve the target accuracy. This reduces computational cost from $O(n^2)$ to $O(m^2)$.

(ii) Smoothness of $\widehat{\phi}_k$: in order to identify the 'smooth' transition of the voting patterns (change points).

Compressive Rank-Polynomial-Basis Design

`Step 1`. Construct the basis functions by Gram Schmidt orthonormalization of powers of $T_1(x;\mathcal{G})$, given by

$$T_1(x;\mathcal{G}) = \frac{\sqrt{12}[F^{\mathrm{mid}}(x;\mathcal{G}) - .5]}{\sqrt{1-\sum_x p^3(x;\mathcal{G})}}, \tag{11}$$

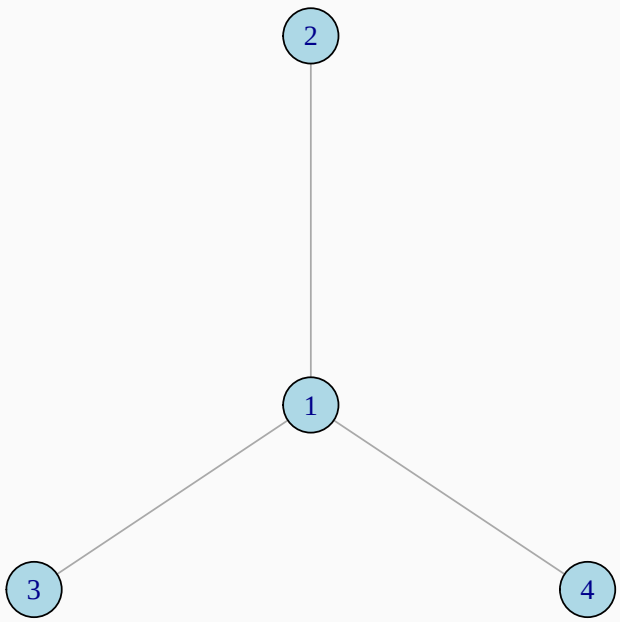
where $F^{\mathrm{mid}}(x;\mathcal{G}) = F(x;\mathcal{G}) - .5p(x;\mathcal{G})$ is the mid-distribution function.

Verify that: These basis functions are orthonormal w.r.t measure F:

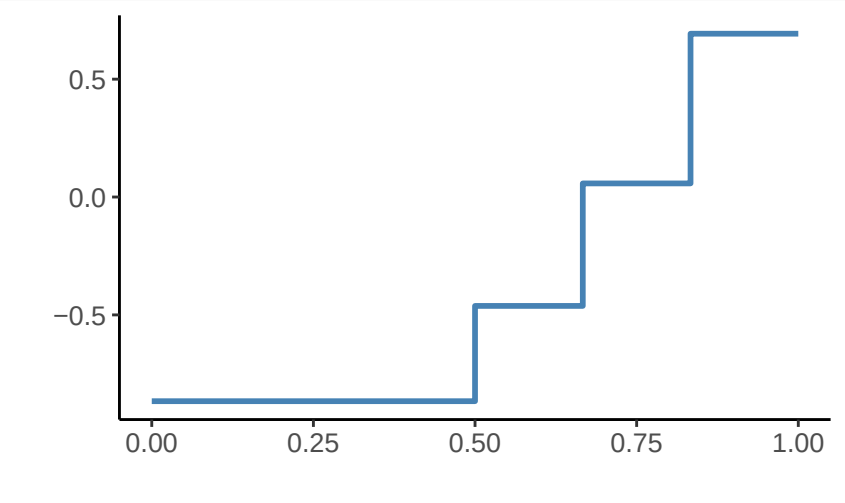
$$\sum_i T_j(x_i;\mathcal{G})p(x_i;\mathcal{G}) = 0, \quad \sum_i T_j(x_i;\mathcal{G})T_k(x_i;\mathcal{G})p(x_i;\mathcal{G}) = \delta_{jk}. \tag{12}$$

`Step 2`. Construct the orthonormal bases in the *unit interval* by evaluating T_j at the quantile function $Q(u;X)$

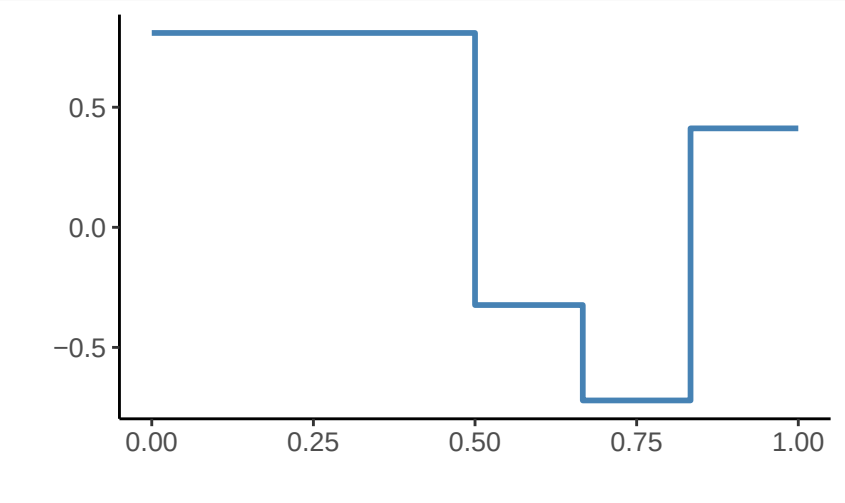
$$\text{LP-Basis}: \quad \eta_j \equiv S_j(u;\mathcal{G}) = T_j(Q(u;X);\mathcal{G}), \quad 0 < u < 1 \tag{13}$$



(a) $S_1(u)$



(b) $S_2(u)$



The LP-Mechanics of Spectral Graph Analysis

$$\text{Master Equation}: \ \mathcal{M}\Theta = H\Theta\Delta.$$

• LP-transform coding of graphs: The **G-matrix** $\mathcal{M}$ with respect to an LP-orthonormal system:

$$\mathrm{LP}[j,k;\mathcal{G}] \quad = \quad \left\langle S_j, \int_0^1 \mathscr{C}\, S_k \right\rangle_{L^2[0,1]} \quad (j,k=1,\ldots,m).$$

• $H_{jl} = \langle S_j, S_l \rangle_{\mathscr{L}^2[0,1]} = \delta_{jl}$ (due to the orthonormality of S_j's).

• Thus we need to perform SVD on the smaller-dimensional $\mathrm{LP}_{m\times m}$ matrix instead of operating on the $n \times n$ dimensional Laplacian matrix, thereby significantly accelerating the computation.

$$\text{LP-Spectral Domain Analysis}: \ \mathrm{LP}\,\Theta = \Theta\Delta.$$

Algorithm 2: An Enhanced Method for Accelerated Graph-Learning

---

`Step 1.` Construct the piecewise-constant orthonormal LP-graph polynomial basis $\{S_1(u;\mathcal{G}), \ldots, S_m(u;\mathcal{G})\}$ using the previous recipe:

`Step 2.` Compute $m \times m$ LP-graph transform matrix (the G-matrix):

$$\mathrm{LP}[j,k;\mathcal{G}] = \sum_{x,y\in V(\mathcal{G})}\sum \widetilde{p}(x,y;\mathcal{G})\, S_j[\widetilde{F}(x;\mathcal{G})]\, S_k[\widetilde{F}(y;\mathcal{G})], \quad 1 \le j,k \le m.$$

`Step 3.` Perform the SVD of LP $= U_{\mathrm{LP}}\Lambda U_{\mathrm{LP}}^T = \sum_k u_k \mu_k u_k^T$, where u_{ij} are the elements of the singular vector of moment matrix $U_{\mathrm{LP}} = (u_1, \ldots, u_m)$, and $\Lambda = \mathrm{diag}(\mu_1, \ldots, \mu_m)$, $\mu_1 \ge \cdots \mu_m \ge 0$.

`Step 4.` Obtain the LP-smoothed graph-Fourier basis by

$$\widehat{\Phi} \leftarrow S \cdot U_{\mathrm{LP}}$$

These "generalized graph-coordinates" are now ready be used for subsequent learning.

LP-Spectral Analysis of Senate Data

```
> library("LPGraph")
> LPSpectral(A,m=15)
```

- K-means clustering on the **smooth** $\widehat{\phi}_1$ produces two groups with the change point at January 19th, 1995.
- Around that time Republicans took control the US House of Representatives for the first time since 1956 and political polarization reached historically high.
- It achieves a remarkable compression ratio of $n/m = 178$, which reduces the memory footprint.
- It delivers **350x** speedup compared to traditional Laplacian method!

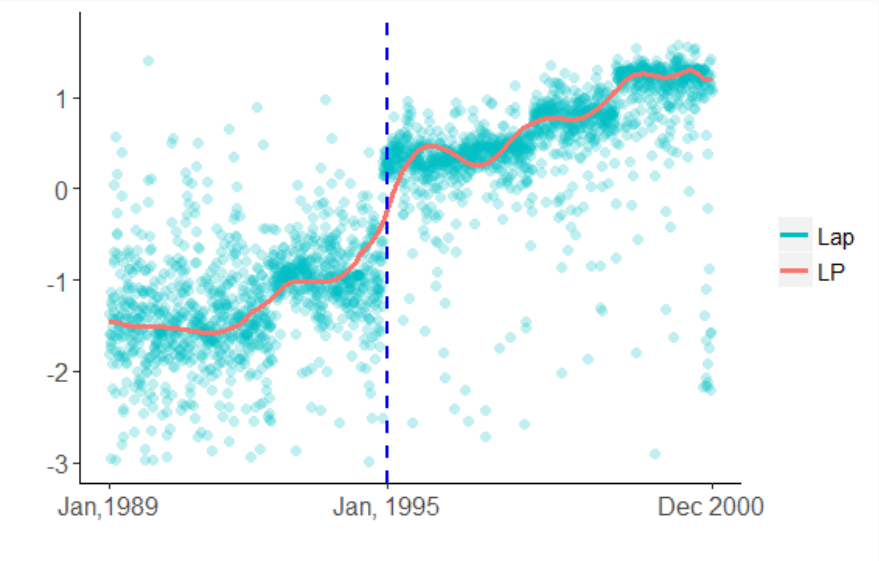


Figure: The 'noisy' blue dots denote the Laplacian eigenmap, and the 'smooth' red line is the LP-spectral embedding.

LPGraph R-Package

LPGraph: Nonparametric Smoothing of Laplacian Graph Spectra

A nonparametric method to approximate Laplacian graph spectra of a network with ordered vertices. This provides a computationally efficient algorithm for obtaining an accurate and smooth estimate of the graph Laplacian basis. The approximation results can then be used for tasks like change point detection, k-sample testing, and so on. The primary reference is Mukhopadhyay, S. and Wang, K. (2018, Technical Report).

Version:	2.0
Depends:	R (≥ 2.10), stats, SDMTools, car, PMA
Published:	2018-05-13
Author:	Subhadeep Mukhopadhyay, Kaijun Wang
Maintainer:	Kaijun Wang <kaijun.wang at temple.edu>
License:	GPL-2
NeedsCompilation:	no
CRAN checks:	LPGraph results

Downloads:

Reference manual:	LPGraph.pdf
Package source:	LPGraph_2.0.tar.gz
Windows binaries:	r-devel: LPGraph_2.0.zip, r-release: LPGraph_2.0.zip, r-oldrel: LPGraph_2.0.zip
OS X binaries:	r-release: LPGraph_2.0.tgz, r-oldrel: LPGraph_2.0.tgz
Old sources:	LPGraph archive

Linking:

Please use the canonical form `https://CRAN.R-project.org/package=LPGraph` to link to this page.

CONCLUSION

The High-Order Bits: “Algorithm of Algorithms”

(1) The prescribed approach accomplishes the miracle of unifying and generalizing the existing paradigm by purely statistical means.

(2) This also provides surprising insights into the design of fast and scalable algorithms for large graphs, which are otherwise hard to guess using previous understanding.

(3) Finally, in the broader context, it allows to bridge the gap between the two modeling cultures:

-*Statistical* (based on nonparametric function approximation and smoothing methods)

-*Algorithmic* (based on matrix theory and numerical linear algebra based techniques).

Selected References

1. Chung, F. R. (1997) *Spectral graph theory*, vol. 92, CBMS Regional Conference Series in Mathematics (American Mathematical Society, USA).

2. Fiedler, M. (1973) Algebraic connectivity of graphs. *Czechoslov. Math. J.* 23, 298–305.

3. Galerkin, B. (1915) Series development for some cases of equilibrium of plates and beams (in Russian). *Wjestnik Ingenerow Petrograd* 19, 897–908.

4. Moody, J. & Mucha, P. J. (2013) Portrait of political party polarization. *Netw. Sci.* 1, 119–121.

5. Parzen, E. (1967) The role of spectral analysis in time series analysis. *Rev. Int. Stat. Inst.* 125–141.

6. Wiener, N. (1930) Generalized harmonic analysis. *Acta mathematica* 55, 117–258.

APPENDIX: OTHER PRACTICAL CONSIDERATIONS

A1. Determining number of clusters k: Senate Data

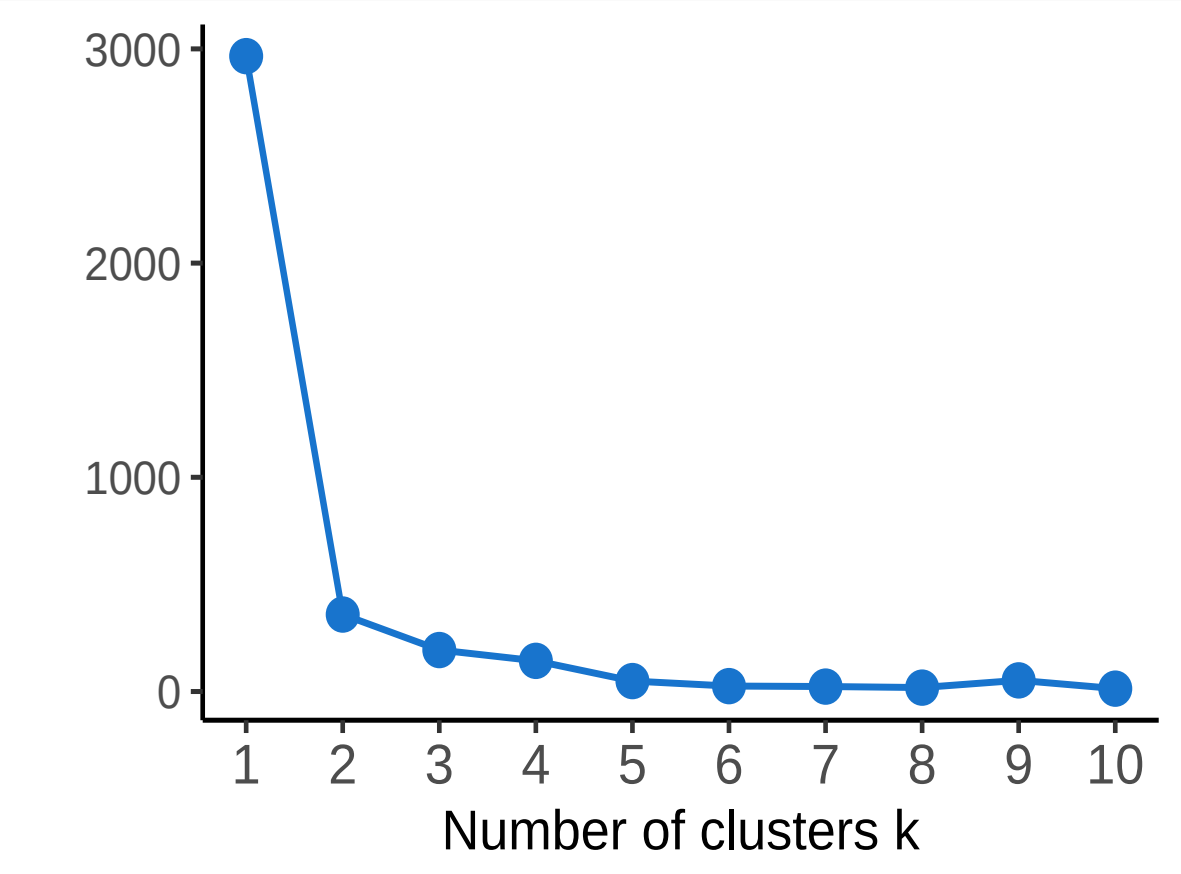


Figure: The Elbow plot: Total within-cluster sum of square for kmeans vs number of clusters for Senate data, which indicates $k = 2$ is an adequate choice. This can be easily computed using the R-function `fviz_nbclust()` available in the package `factoextra`.

A2. Additional Change Point Examples

- Compare the statistical efficiency of LP and Laplacian approaches using location-scale model:

$$X_t = \begin{cases} H_t, & 1 \leq t \leq \frac{n}{2} \\ \mu \cdot \mathbf{1}_d + s \cdot Z_t, & \frac{n}{2} < t \leq n \end{cases} \tag{14}$$

- $Z_t \sim \mathcal{N}_d(\mathbf{0}_d, \boldsymbol{I}_d)$; $\mathbf{1}_d = (1, 1, \ldots, 1)$; $\mu = 0.3$; $s = 1.2$; $d = 500$
- Comparisons based on three different choices of n: 500, 1000 and 2500.
- Two cases for each n:
  - Case 1: $H_t = Z_t$;
  - Case 2: $H_t = T_d(\mathbf{0}_d, \boldsymbol{I}_d)$ with degrees of freedom 3.
- Each case is repeated for 250 time.

Statistical Efficiency-Comparison Chart

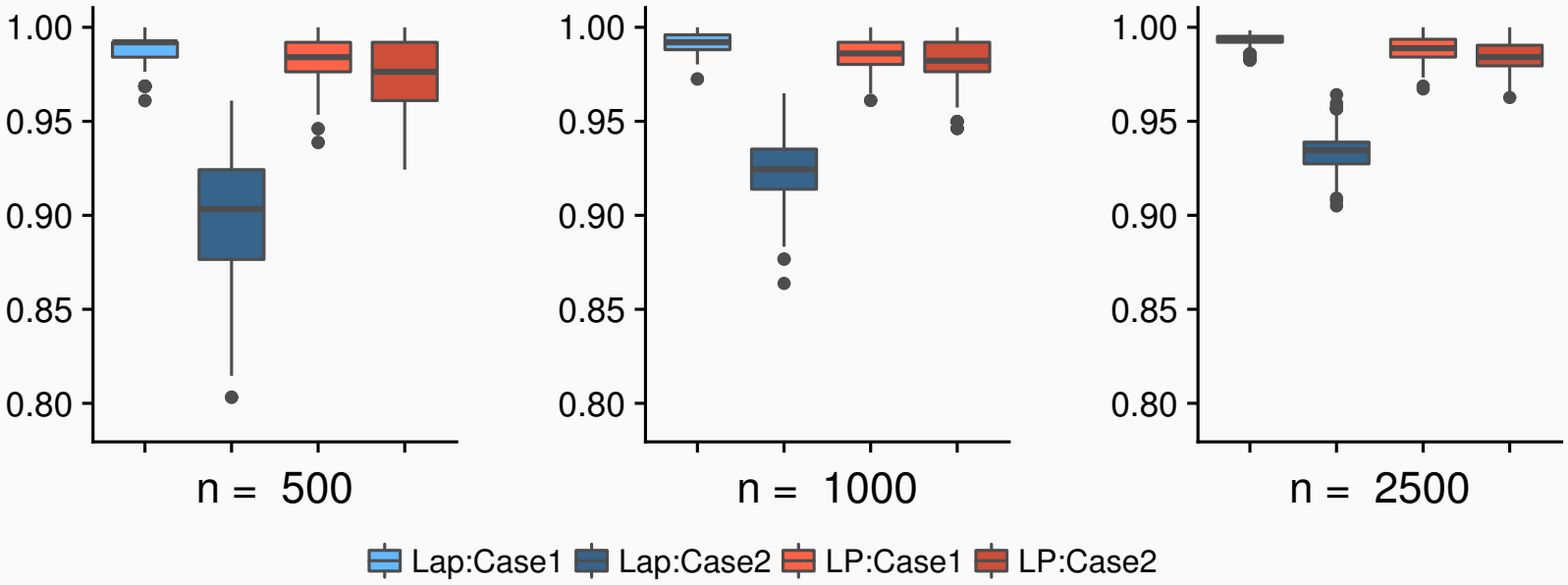


Figure: Statistical Efficiency: Jaccard criteria boxplots comparison for Laplacian spectra and LP approximation.

Computational Efficiency

Table: Computation efficiency: relative computation speed ($\mathrm{time}_{\mathrm{Lap}}/\mathrm{time}_{\mathrm{LP}}$)

n	500	1000	2500
Relative Speed Gain	23.78	115.44	333.16

- LP-spectral runs more than 300x faster than the traditional Laplacian-based method for moderately large-sized graphs.

A3. Visualizing GraField

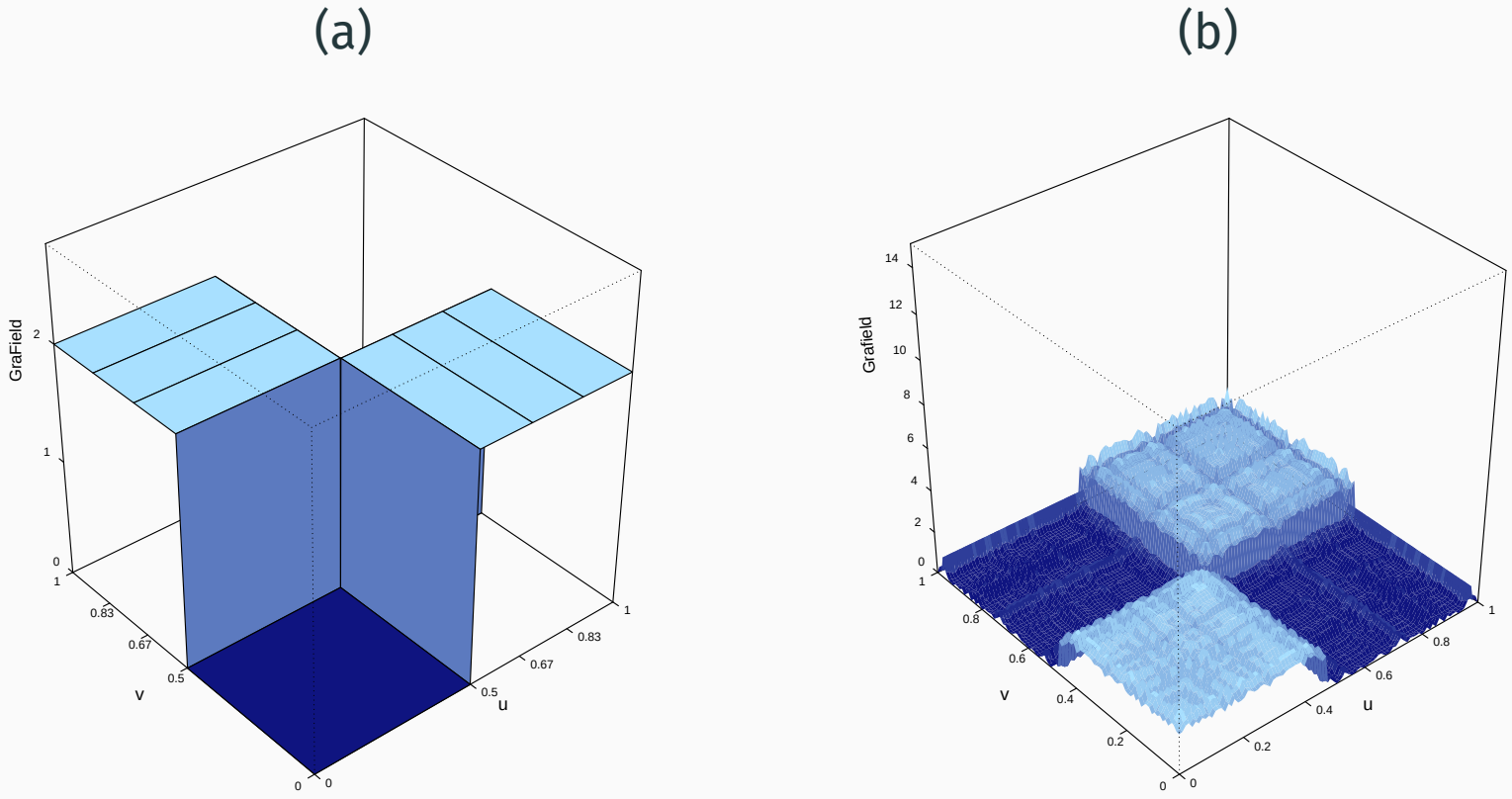


Figure: GraField function for (a) 3-star network ; (b) US political weblogs data (approximated).