Signal Processing … aphs

SpaRTaN-MacS … School

Low-Pass Filtering Strikes Back!

Pierre Vandergheynst

…wiss Federal Institute of Technology

April is Autism Awareness Month: https://www.autismspeaks.org/wordpress-tags/autism-awareness-month

LTS EPFL

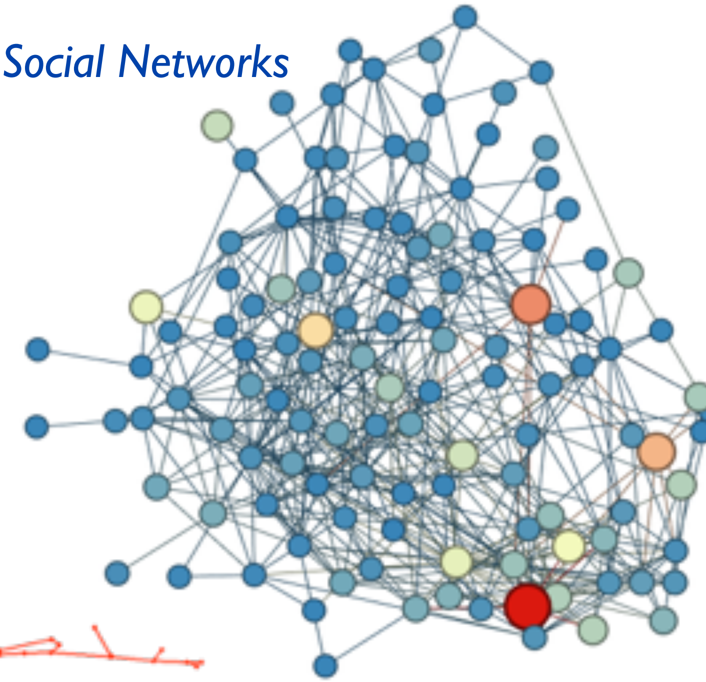
LTS2

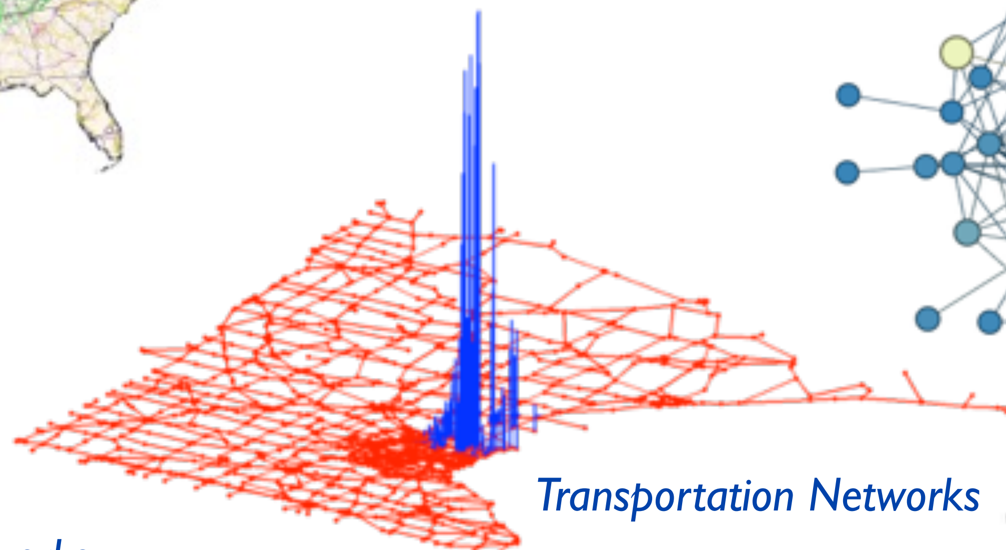
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Signal Processing on Graphs

Energy Networks

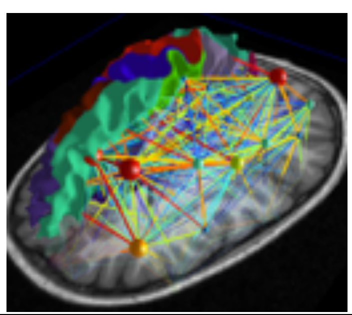
Social Networks

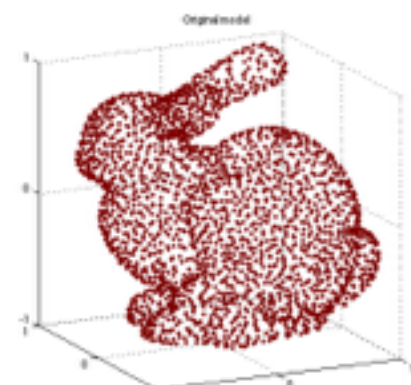




Transportation Networks

Biological Networks



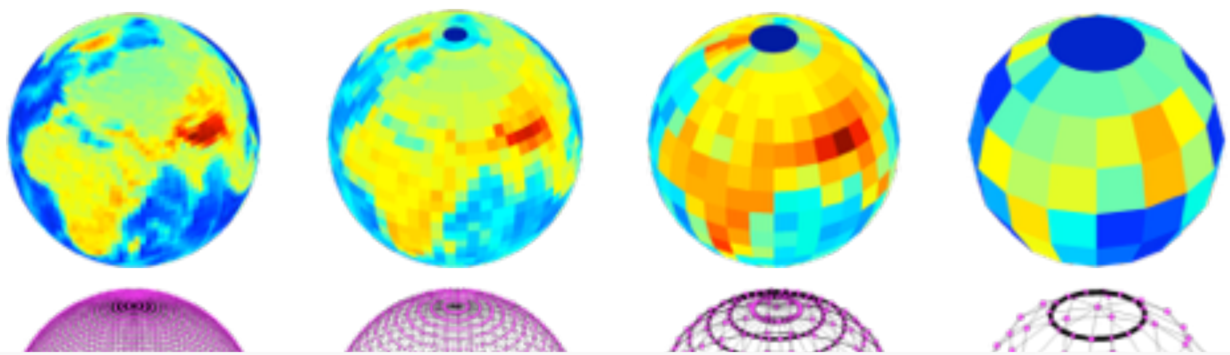


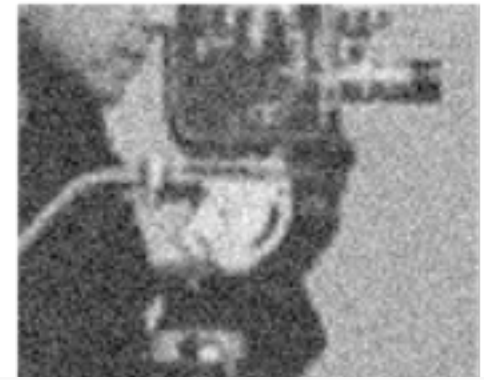
Irregular Data Domains


330
Accounts Created
171000
Tweets
YouTube
69420
Video Hours
Watched
60
Video Hours
Uploaded
173610
+1s
690
Blog Posts
Google
138240
Searches
$48060
Ad Revenue
amazon
1530
Items Purchased
$70770
Money Spent
foursquare
1050
Check-Ins
yelp
15
Reviews
19020
App Downloads
37080
App Downloads
1565880 Likes
1649280 Posts
180 GB of Data

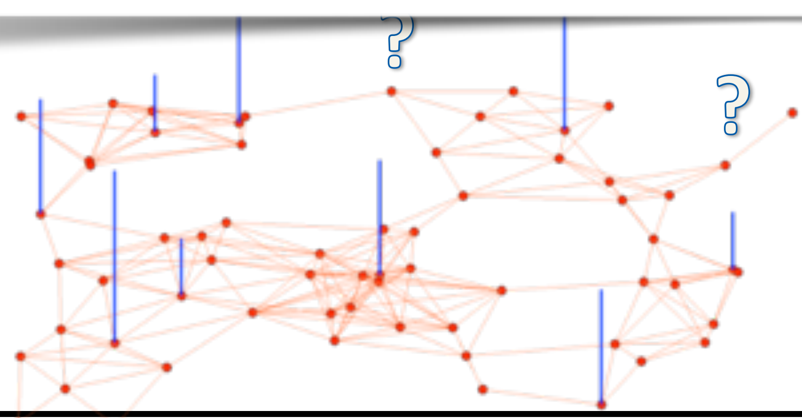
Some Typical Processing Problems

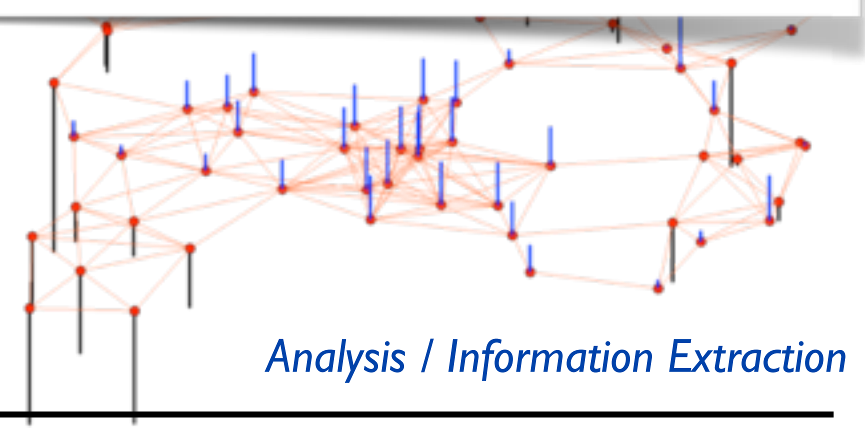
Compression / Visualization





Many interesting new contributions with a SP perspective
[Coifman, Maggioni, Kolaczyk, Ortega, Ramchandran, Moura, Lu, Borgnat]
or IP perspective [ElMoataz, Lezoray]
See review in 2013 IEEE SP Mag





Analysis / Information Extraction

Outline

- Introduction:
  - Graphs and elements of spectral graph theory, with emphasis on functional calculs
- Kernel Convolution:
  - Localization, filtering, smoothing and applications
- An application to spectral clustering that unifies some of the themes you've heard of during the workshop: machine learning, compressive sensing, optimisation algorithms, graphs

Elements of Spectral Graph Theory

Reference: F. Chung, Spectral Graph Theory

Definitions

A graph G is given by a set of vertices and «relationships » between them encoded in edges G $= (V,E)$

A set V of vertices of cardinality $|V| = N$

A set E of edges: $e \in E, \quad e = (u, v)$ with $u, v \in V$

Directed edge: $e = (u, v), \quad e' = (v, u)$ and $e \neq e'$

Undirected edge: $e = (u, v), \quad e' = (v, u)$ and $e = e'$

A graph is undirected if it contains only undirected edges

A weighted graph has an associated non-negative weight function:

$$w : V \times V \to \mathbb{R}^+ \quad (u, v) \notin E \Rightarrow w(u, v) = 0$$

Matrix Formulation

Connectivity captured via the (weighted) adjacency matrix

$$W(u,v) = w(u,v)$$ with obvious restriction for unweighted graphs

$$W(u,u) = 0$$ no loops

Let $d(u)$ be the degree of u and $\mathbf{D} = \mathrm{diag(d)}$ the degree matrix

Graph Laplacians, Signals on Graphs

$$\mathcal{L} = \mathbf{D} - \mathbf{W} \qquad \mathcal{L}_{\text{norm}} = \mathbf{D}^{-1/2}\mathcal{L}\mathbf{D}^{-1/2}$$

Graph signal: $f : V \to \mathbb{R}$

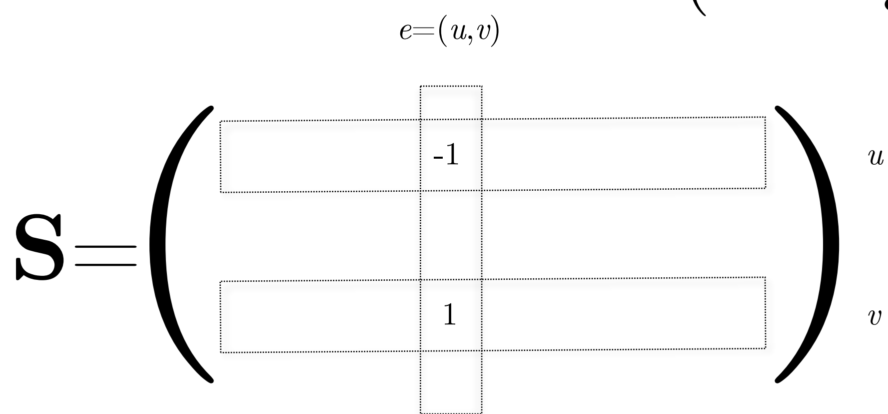
Laplacian as an operator on space of graph signals

$$\mathcal{L}f(u) = \sum_{v \sim u} w(u,v)\big(f(u) - f(v)\big)$$

Some differential operators

The Laplacian can be factorized as $\mathcal{L} = \mathbf{S}\mathbf{S}^*$

Explicit form of the incidence matrix (unweighted in this example):



$$\mathbf{S}^* f(u,v) = f(v) - f(u)$$ is a gradient

$$\mathbf{S}g(u) = \sum_{(u,v)\in E} g(u,v) - \sum_{(v',u)\in E} g(v',u)$$ is a negative divergence

Properties of the Laplacian

Laplacian is symmetric and has real eigenvalues

Moreover: $\langle f, \mathcal{L} f \rangle = \sum_{u \sim v} w(u,v)\big(f(u) - f(v)\big)^2 \geq 0$ Dirichlet form

positive semi-definite, non-negative eigenvalues

Spectrum: $0 = \lambda_0 \leq \lambda_1 \leq \ldots \lambda_{\max}$

G connected: $\lambda_1 > 0$

$\lambda_i = 0$ and $\lambda_{i+1} > 0$ $\quad G$ has $i{+}1$ connected components

Notation: $\langle f, \mathcal{L} g \rangle = f^t \mathcal{L} g$

Measuring Smoothness

$$\langle f, \mathcal{L}f \rangle = \sum_{u \sim v} \big(f(u) - f(v)\big)^2 \geq 0$$

is a measure of « how smooth » f is on G

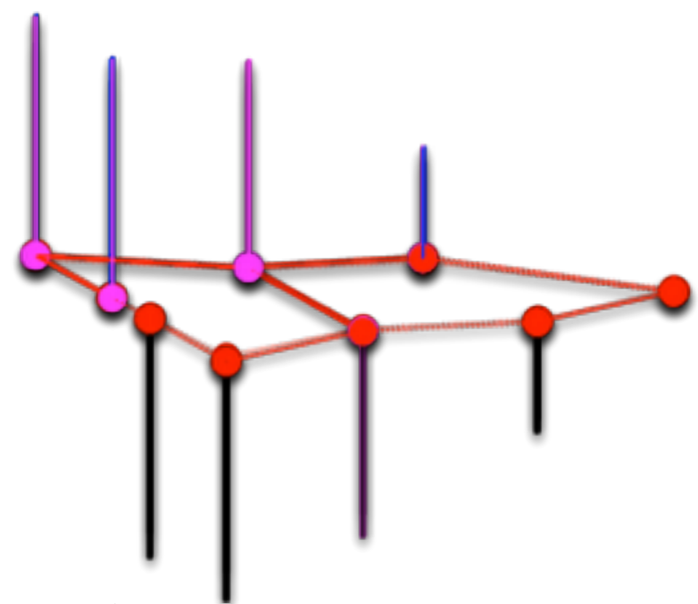
Using our definition of gradient: $\nabla_u f = \{S^* f(u,v), \forall v \sim u\}$

Local variation $\|\nabla_u f\|_2 = \sqrt{\sum_{v \sim u} |S^* f(u,v)|^2}$

Total variation $|f|_{TV} = \sum_{u \in V} \|\nabla_u f\|_2 = \sum_{u \in V} \sqrt{\sum_{v \sim u} |S^* f(u,v)|^2}$

Notions of Global Regularity for Graph

Discrete Calculus, Grady and Polimeni, 2010



Edge Derivative

$$\left.\frac{\partial \mathbf{f}}{\partial e}\right|_m := \sqrt{w(m,n)}\,[f(n) - f(m)]$$

Graph Gradient

$$\nabla_m \mathbf{f} := \left[\left\{\left.\frac{\partial \mathbf{f}}{\partial e}\right|_m\right\}_{e\in\mathcal{E} \text{ s.t. } e=(m,n)}\right]$$

Local Variation

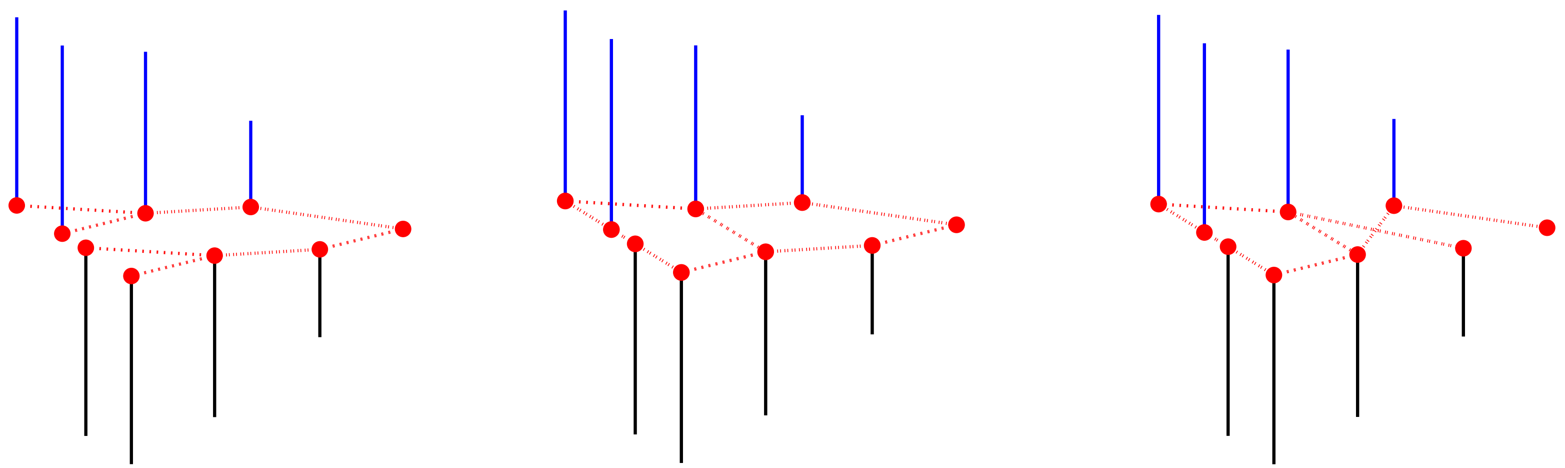
$$||\nabla_m \mathbf{f}||_2 = \left[\sum_{n\in\mathcal{N}_m} w(m,n)\,[f(n) - f(m)]^2\right]^{\frac{1}{2}}$$

Quadratic Form

$$\frac{1}{2}\sum_{m\in V} ||\nabla_m \mathbf{f}||_2^2 = \sum_{(m,n)\in\mathcal{E}} w(m,n)\,[f(n) - f(m)]^2 = \mathbf{f}^{\mathrm{T}}\mathcal{L}\mathbf{f}$$

Smoothness of Graph Signals

$\mathcal{G}_1$ $\mathcal{G}_2$ $\mathcal{G}_3$



$\mathbf{f}^{\mathrm{T}}\mathcal{L}_1\mathbf{f} = 0.14$ $\mathbf{f}^{\mathrm{T}}\mathcal{L}_2\mathbf{f} = 1.31$ $\mathbf{f}^{\mathrm{T}}\mathcal{L}_3\mathbf{f} = 1.81$

Remark on Discrete Calculus

Discrete operators on graphs form the basis of an interesting field aiming at bringing a PDE-like framework for computational analysis on graphs:

- Leo Grady: Discrete Calculus
- Olivier Lezoray, Abderrahim Elmoataz and co-workers: PDEs on graphs:
  - many methods from PDEs in image processing can be transposed on arbitrary graphs
  - applications in vision (point clouds) but also machine learning (inference with graph total variation)

Laplacian eigenvectors

Spectral Theorem: Laplacian is PSD with eigen decomposition

$$\mathcal{L} = \mathbf{D} - \mathbf{W} \qquad \{(\lambda_\ell, \mathbf{u}_\ell)\}_{\ell=0,1,\ldots,N-1}$$

$$\mathcal{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^t$$

That particular basis will play the role of the Fourier basis:

Graph Fourier Transform, Coherence

$$\hat{f}(\lambda_\ell) := \langle \mathbf{f}, \mathbf{u}_\ell \rangle = \sum_{i=1}^{N} f(i) u_\ell^*(i)$$

$$\mu := \max_{\ell,i} |\langle \mathbf{u}_\ell, \delta_i \rangle| \in \left[\frac{1}{\sqrt{N}}, 1 \right[$$

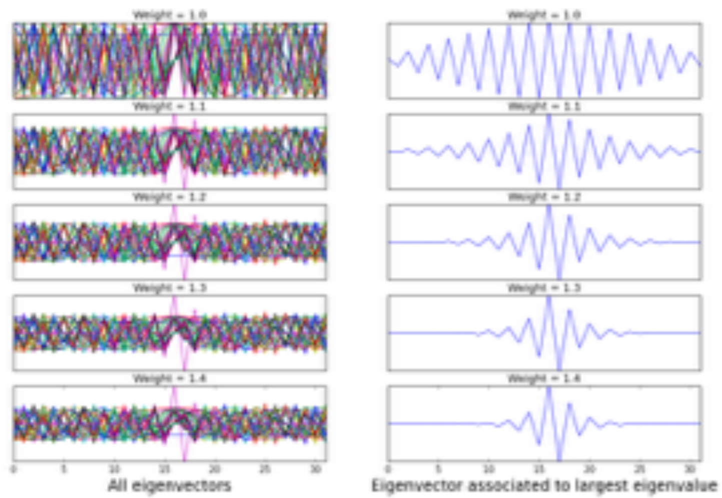
Graph Coherence

Important remark on eigenvectors

$$\mu := \max_{\ell,i} |\langle \mathbf{u}_\ell, \delta_i \rangle| \in \left[\frac{1}{\sqrt{N}}, 1\right]$$

Optimal - Fourier case

What does that mean ??



Eigenvectors of modified path graph

Examples: Cut and Clustering

$$C(A,B) := \sum_{i\in A, j\in B} W[i,j] \quad \mathrm{RatioCut}(A,\overline{A}) := \frac{1}{2}\frac{C(A,\overline{A})}{|A|} + \frac{1}{2}\frac{C(A,\overline{A})}{|\overline{A}|}$$

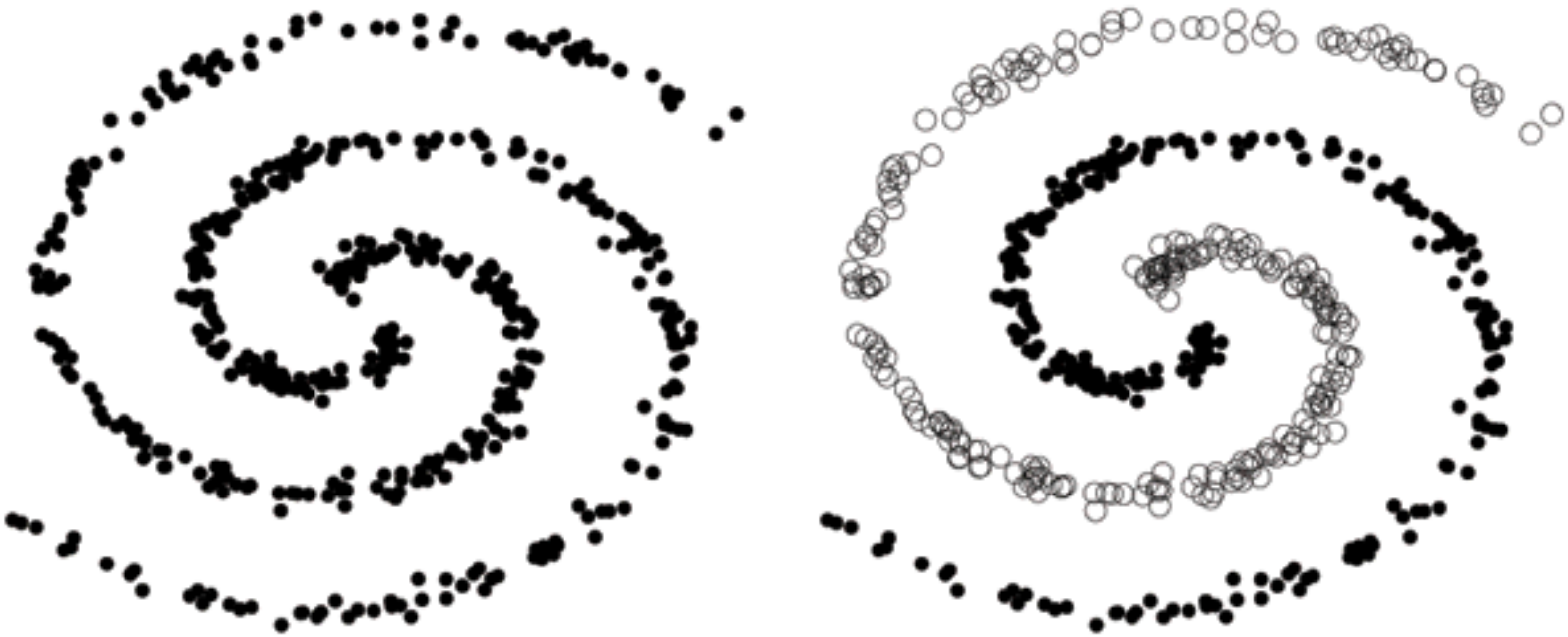
$$\min_{A\subset V} \mathrm{RatioCut}(A,\overline{A}) \qquad f[i] = \begin{cases} \sqrt{|\overline{A}|/|A|} & \text{if } i \in A \\ -\sqrt{|A|/|\overline{A}|} & \text{if } i \in \overline{A} \end{cases}$$

$$\|f\| = \sqrt{|V|} \text{ and } \langle f, 1\rangle = 0$$

$$f^t \mathcal{L} f = |V| \cdot \mathrm{RatioCut}(A,\overline{A})$$

$$\arg\min_{f\in\mathbb{R}^{|V|}} f^t \mathcal{L} f \text{ subject to } \|f\| = \sqrt{|V|} \text{ and } \langle f, 1\rangle = 0$$

Relaxed problem Looking for a smooth partition function



Examples: Cut and Clustering

Spectral Clustering

$$\arg\min_{f \in \mathbb{R}^{|V|}} f^t \mathcal{L} f \text{ subject to } \|f\| = \sqrt{|V|} \text{ and } \langle f, 1 \rangle = 0$$

By Rayleigh-Ritz, solution is second eigenvector $\mathbf{u}_1$

Remarks: Natural extension to more than 2 sets

Solution is real-valued and needs to be quantized.

In general, k-MEANS is used.

First k eigenvectors of sparse Laplacians via Lanczos, complexity driven by eigengap $|\lambda_k - \lambda_{k+1}|$

Spectral clustering := embedding + k-MEANS

$$\forall i \in V : i \mapsto (u_0(i), \dots, u_{k-1}(i))$$

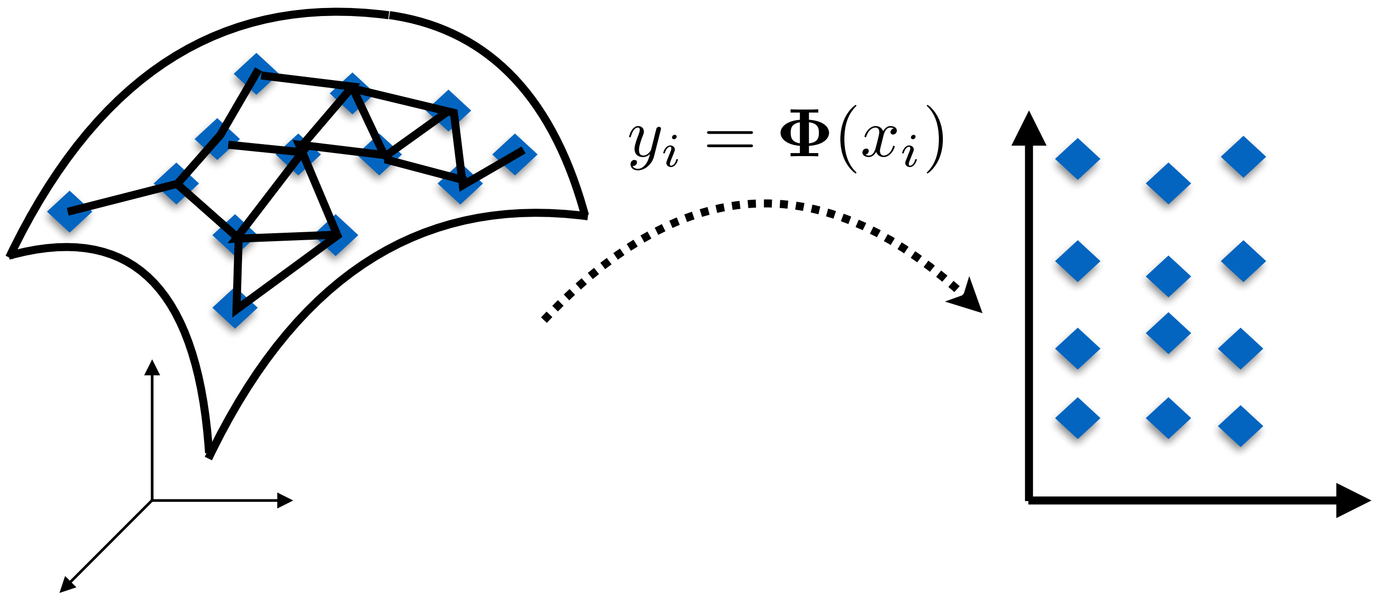
Graph Embedding/Laplacian Eigenmaps

Goal: embed vertices in **low** dimensional space, discovering geometry

$$(x_1, \dots x_N) \mapsto (y_1, \dots y_N)$$

$$x_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}^k \qquad k < d$$

Good embedding: nearby points mapped nearby, **so smooth map**



Graph Embedding/Laplacian Eigenmaps

Goal: embed vertices in **low** dimensional space, discovering geometry

$$(x_1, \dots x_N) \mapsto (y_1, \dots y_N)$$

$$x_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}^k \qquad k < d$$

Good embedding: nearby points mapped nearby, **so smooth map**

minimize variations/
maximize smoothness of embedding $\sum_{i,j} W[i,j](y_i - y_j)^2$

Laplacian Eigenmaps

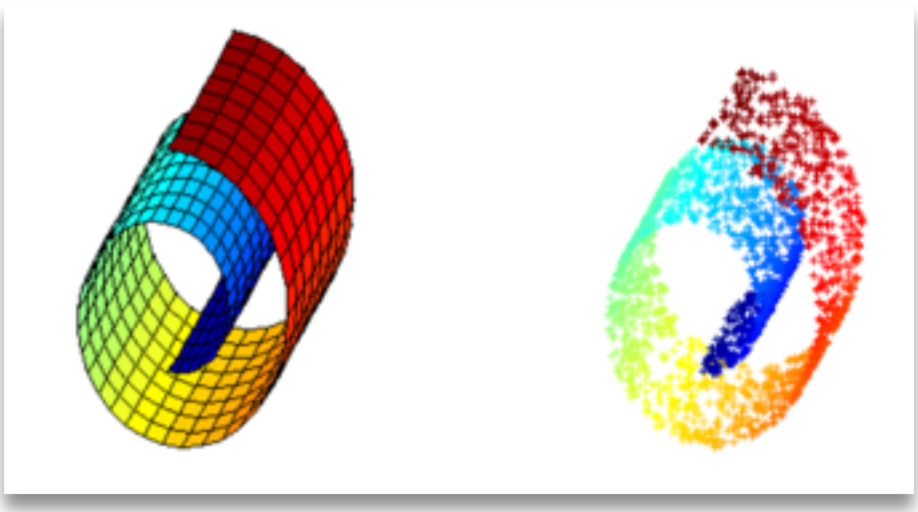
$$\arg \min_{\substack{\mathbf{y} \\ \mathbf{y}^t \mathbf{D} \mathbf{y} = 1 \\ \mathbf{y}^t \mathbf{D} 1 = 0}} \mathbf{y}^t \mathcal{L} \mathbf{y}$$

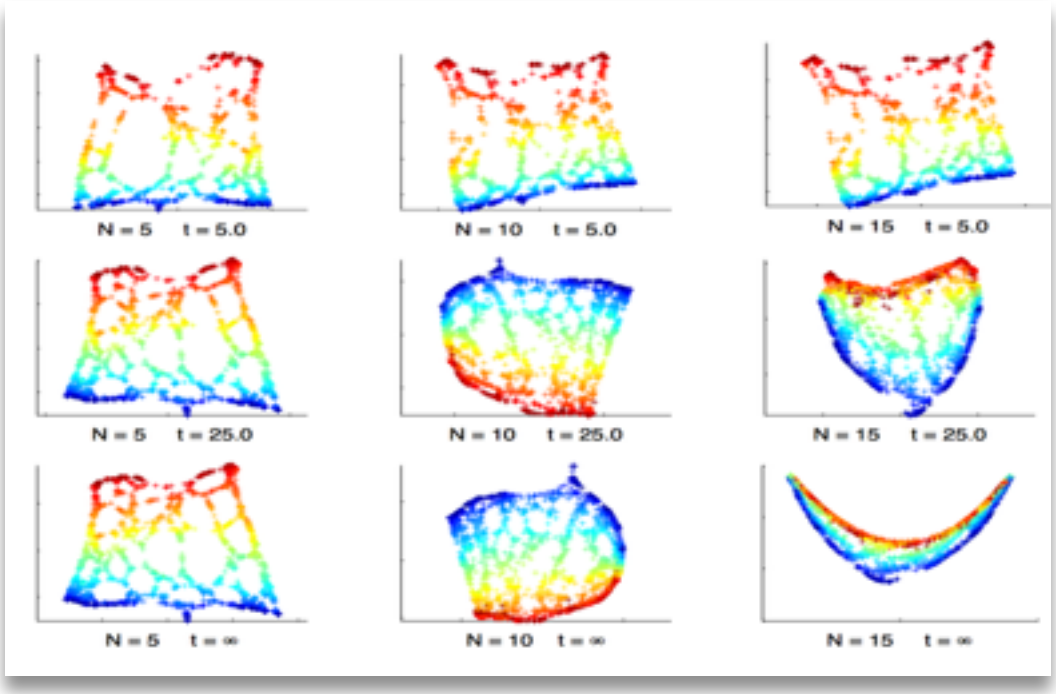
fix scale

$\rightarrow$

$$\mathcal{L}\mathbf{y} = \lambda \mathbf{D} \mathbf{y}$$

Laplacian Eigenmaps





[Belkin, Niyogi, 2003]

Remark on Smoothness

Linear / Sobolev case

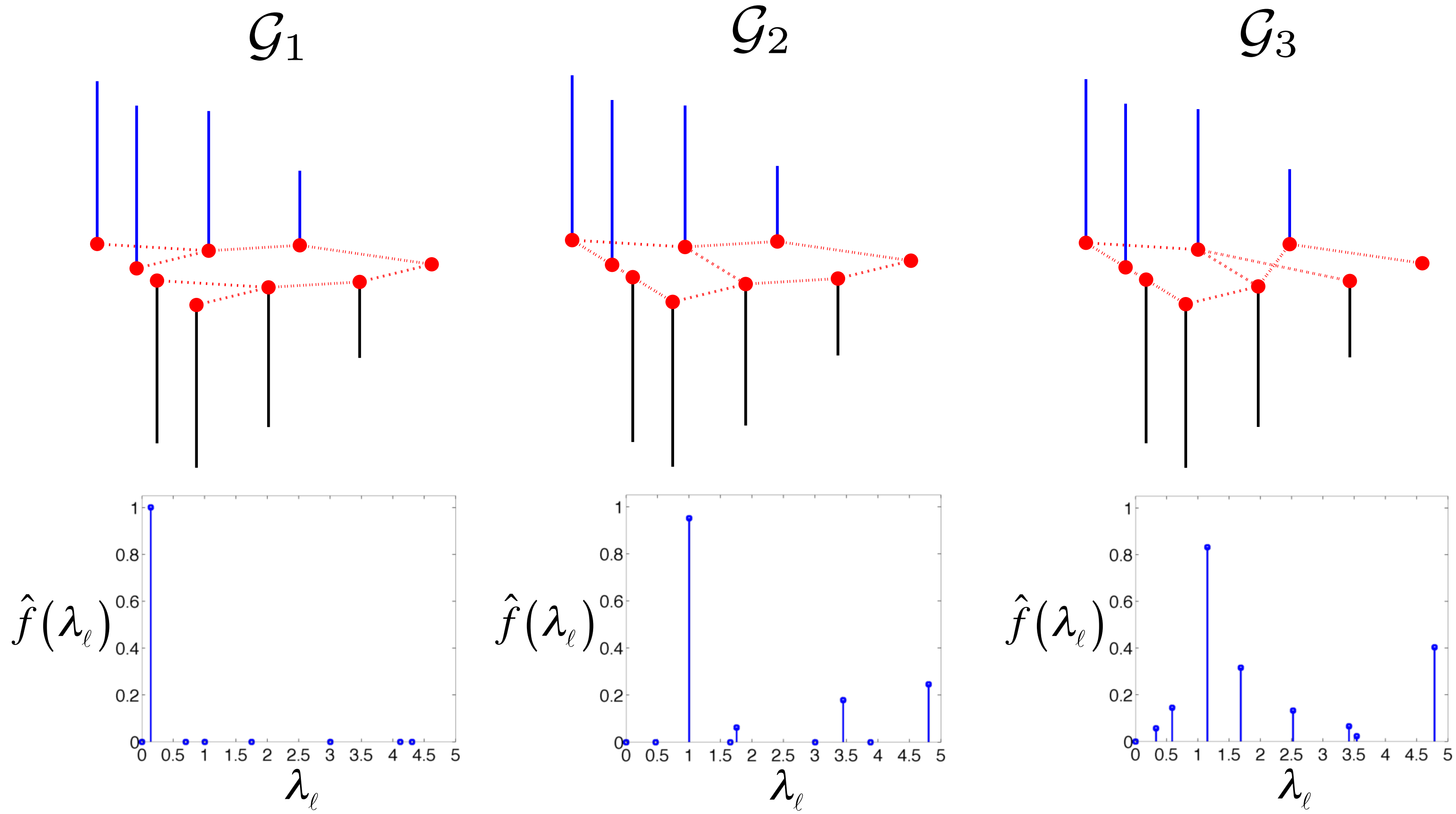
Smoothness, loosely defined, has been used to motivate various methods and algorithms. But in the discrete, finite dimensional case, asymptotic decay does not mean much

$$\|\nabla f\|_2^2 \leq M \Leftrightarrow f^t \mathcal{L} f \leq M \Leftrightarrow \sum_{\ell} \lambda_\ell |\hat{f}(\ell)|^2 \leq M$$

$$E_K(f) = \|f - P_K(f)\|_2 \qquad E_K(f) \leq \frac{\|\nabla f\|_2}{\sqrt{\lambda_{K+1}}}$$

$$|\hat{f}(\ell)| \leq \frac{\sqrt{M}}{\sqrt{\lambda_\ell}}$$

Smoothness of Graph Signals Revisited



$\mathbf{f}^{\mathrm{T}}\mathcal{L}_1\mathbf{f} = 0.14$ $\mathbf{f}^{\mathrm{T}}\mathcal{L}_2\mathbf{f} = 1.31$ $\mathbf{f}^{\mathrm{T}}\mathcal{L}_3\mathbf{f} = 1.81$

Functional calculus

It will be useful to manipulate functions of the Laplacian

$$f(\mathcal{L}),\ f : \mathbb{R} \mapsto \mathbb{R}$$

$\mathcal{L}^k \mathbf{u}_\ell = \lambda_\ell^k \mathbf{u}_\ell$ ⟶ polynomials

Symmetric matrices admit a (Borel) functional calculus

Borel functional calculus for symmetric matrices

$$f(\mathcal{L}) = \sum_{\ell \in \mathcal{S}(\mathcal{L})} f(\lambda_\ell) \mathbf{u}_\ell \mathbf{u}_\ell^t$$

Use spectral theorem on powers, get to polynomials

From polynomial to continuous functions by Stone-Weierstrass

Then Riesz-Markov (non-trivial !)

Example: Diffusion on Graphs

Consider the following « heat » diffusion model

$$\frac{\partial f}{\partial t} = -\mathcal{L} f \qquad \frac{\partial}{\partial t}\hat{f}(\ell, t) = -\lambda_\ell \hat{f}(\ell, t) \qquad \hat{f}(\ell, 0) := \hat{f}_0(\ell)$$

$$\hat{f}(\ell, t) = e^{-t\lambda_\ell}\hat{f}_0(\ell) \qquad f = e^{-t\mathcal{L}} f_0$$ by functional calculus

Explicitly:

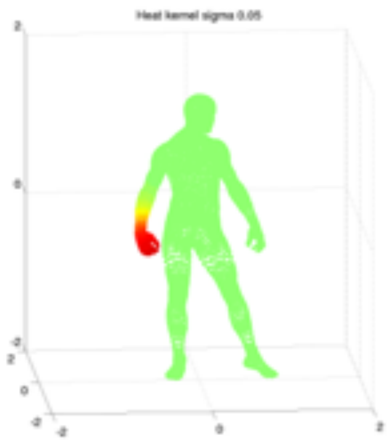
$$f(i) = \sum_{j \in V}\sum_\ell e^{-t\lambda_\ell} u_\ell(i) u_\ell(j) f_0(j)$$

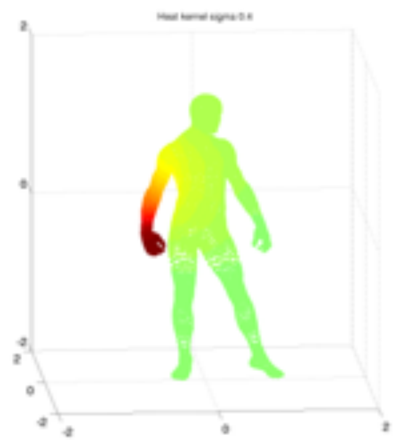
$$e^{-t\mathcal{L}} = \sum_\ell e^{-t\lambda_\ell}\mathbf{u}_\ell \mathbf{u}_\ell^t \qquad = \sum_\ell e^{-t\lambda_\ell} u_\ell(i) \sum_{j \in V} u_\ell(j) f_0(j)$$

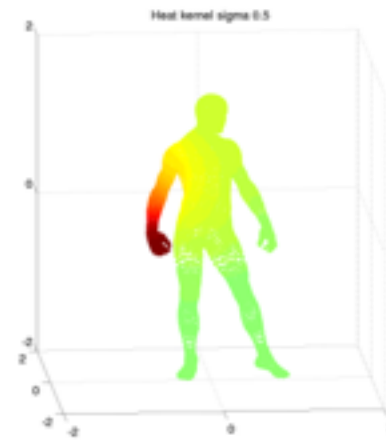
$$e^{-t\mathcal{L}}[i, j] = \sum_\ell e^{-t\lambda_\ell} u_\ell(i) u_\ell(j) \qquad = \sum_\ell e^{-t\lambda_\ell}\hat{f}_0(\ell) u_\ell(i)$$

Example: Diffusion on Graphs

examples of heat kernel on graph



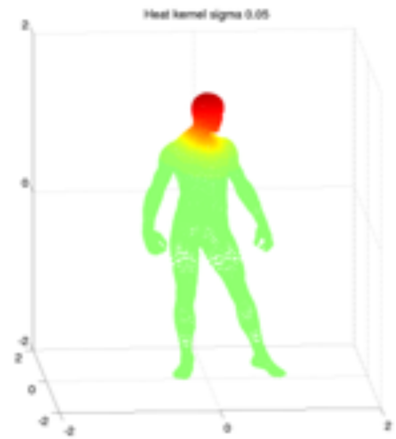




$$f_0(j) = \delta_k(j)$$

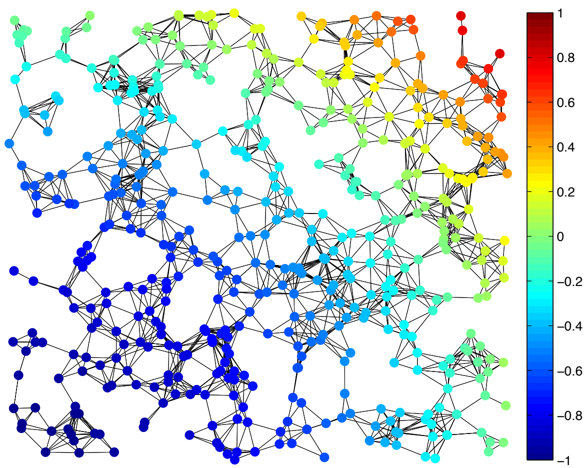
$$f(i) = \sum_{\ell} e^{-t\lambda_\ell} \hat{f}_0(\ell) u_\ell(i)$$

$$= \sum_{\ell} e^{-t\lambda_\ell} u_\ell(k) u_\ell(i)$$



Simple De-Noising Example

Suppose a smooth signal f on a graph



Original

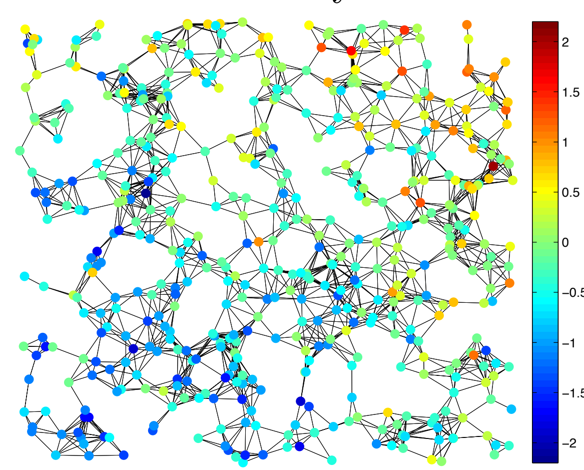
$$\|\nabla f\|_2^2 \leq M \Leftrightarrow f^t \mathcal{L} f \leq M$$

$$|\hat{f}(\ell)| \leq \frac{\sqrt{M}}{\sqrt{\lambda_\ell}}$$

But you observe only a noisy version y

$$y(i) = f(i) + n(i)$$

Noisy



Simple De-Noising Example

De-Noising by Regularization

$$\underset{f}{\operatorname{argmin}} \|f - y\|_2^2 \text{ s.t. } f^t \mathcal{L} f \leq M$$

$$\underset{f}{\operatorname{argmin}} \frac{\tau}{2}\|f - y\|_2^2 + f^{\mathrm{T}} \mathcal{L}^r f \quad \Rightarrow \quad \mathcal{L}^r f_* + \frac{\tau}{2}(f_* - y) = 0$$

Graph Fourier

$$\Rightarrow \quad \widehat{\mathcal{L}^r f_*}(\ell) + \tfrac{\tau}{2}\left(\widehat{f_*}(\ell) - \hat{y}(\ell)\right) = 0, \quad \forall \ell \in \{0, 1, \ldots, N-1\}$$

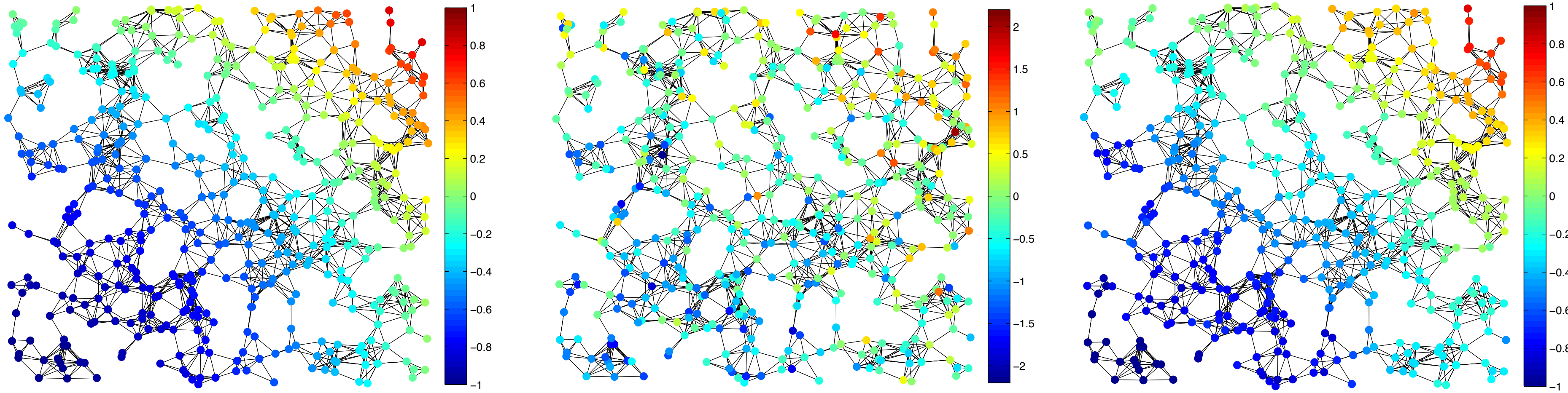
$$\Rightarrow \quad \widehat{f_*}(\ell) = \frac{\tau}{\tau + 2\lambda_\ell^r} \hat{y}(\ell)$$

"Low pass" filtering !

Convolution with a kernel: $\hat{f}(\ell)\hat{g}(\lambda_\ell; \tau, r) \Rightarrow g(\mathcal{L}; \tau, r)$

Simple De-Noising Example

$$\text{argmin}_f \left\{ ||f - y||_2^2 + \gamma f^T \mathcal{L} f \right\}$$



Original Noisy Denoised

$$\underset{f}{\text{argmin}} \frac{\tau}{2}\|f - y\|_2^2 + f^{\mathrm{T}}\mathcal{L}^r f \quad \Rightarrow \quad \mathcal{L}^r f_* + \frac{\tau}{2}(f_* - y) = 0$$

$$\Rightarrow \quad \widehat{f_*}(\ell) = \frac{\tau}{\tau + 2\lambda_\ell^r}\hat{y}(\ell)$$ "Low pass" filtering !

Filtering: $\hat{f}_{out}(\lambda_\ell) = \hat{f}_{in}(\lambda_\ell)\hat{h}(\lambda_\ell)$ $\quad f_{out}(i) = \sum_{\ell=0}^{N-1} \hat{f}_{in}(\lambda_\ell)\hat{h}(\lambda_\ell)u_\ell(i)$

Convolution with a kernel and localization

"Convolutions" and "Translations"

$$(f * g)(n) = \sum_{\ell} \hat{f}(\ell)\hat{g}(\ell)u_\ell(n)$$

Inherits a lot of properties of the usual convolution

associativity, distributivity, diagonalized by GFT

$$g_0(n) := \sum_{\ell} u_\ell(n) \quad \Longrightarrow \quad f * g_0 = f$$

$$\mathcal{L}(f * g) = (\mathcal{L}f) * g = f * (\mathcal{L}g)$$

Use convolution to induce translations

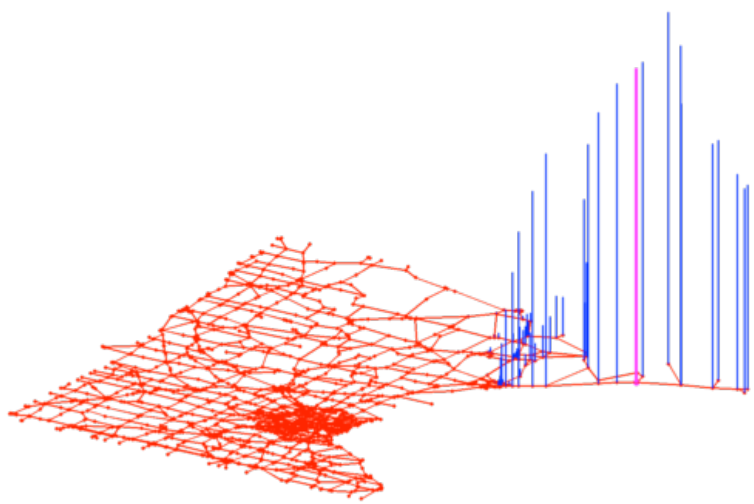
$$(T_i f)(n) := \sqrt{N}(f * \delta_i)(n) = \sqrt{N} \sum_{\ell} \hat{f}(\ell) u_\ell^*(i) u_\ell(n)$$

Localising a Kernel

Hammond et al., Wavelets on graphs via spectral graph theory, ACHA, 2011

- Action of the localisation operator on a spectral kernel

$$\big(T_i f\big)(n) := \sqrt{N}(f * \delta_i)(n) = \sqrt{N}\sum_{\ell} \hat{f}(\ell) u_\ell^*(i) u_\ell(n)$$



The Agonizing Limits of Intuition

The Graph Fourier and Kronecker bases are not necessarily mutually unbiased

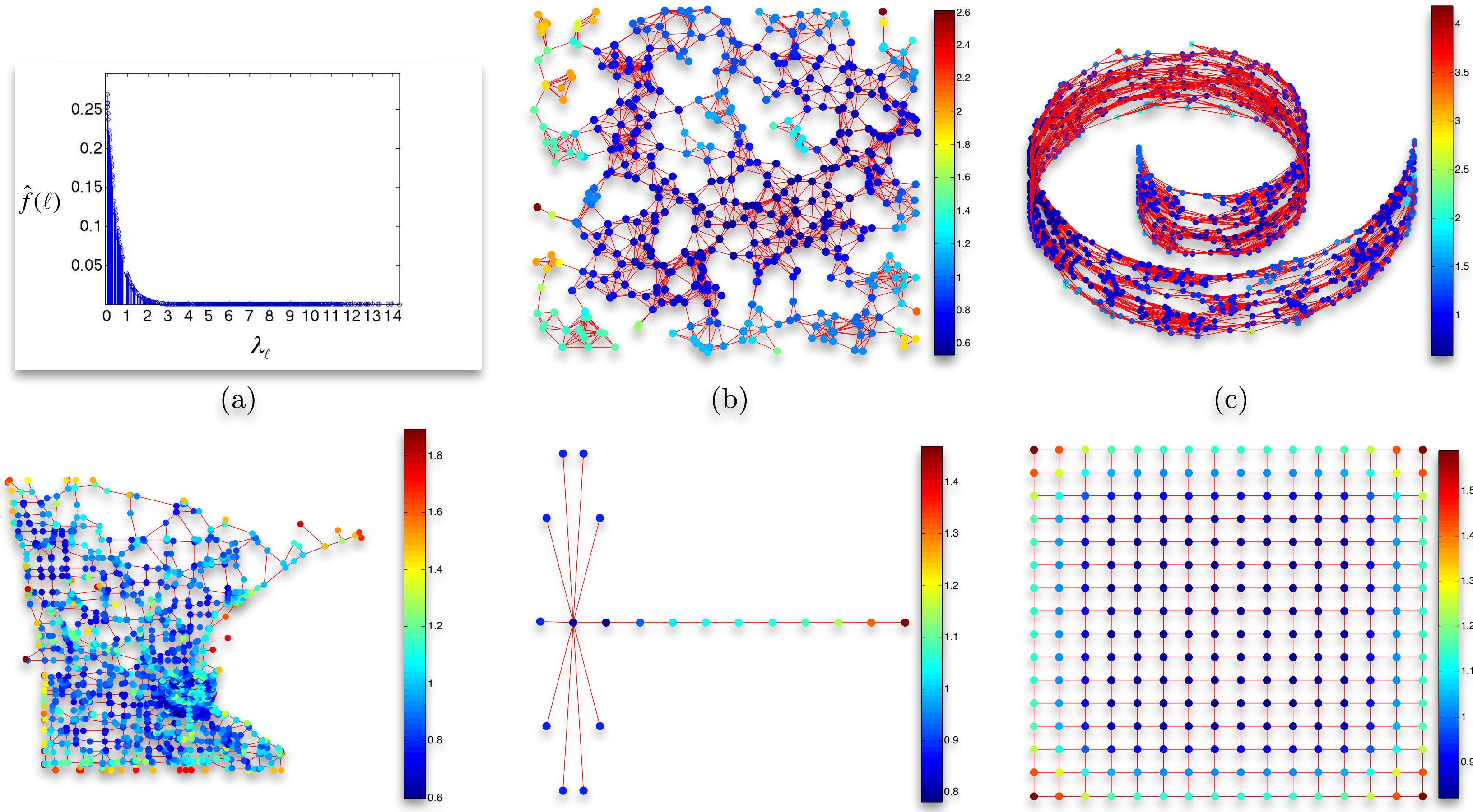
$$\mu := \max_{\ell,i} |\langle \mathbf{u}_\ell, \delta_i \rangle| \in \left[\frac{1}{\sqrt{N}}, 1 \right[$$

Laplacian eigenvectors (Fourier modes!) can be well localized

- phenomenon not yet fully understood, under intense study
- can be observed in lots of experimental data graphs
- not universal: known classes of random and regular graphs have delocalized eigenvectors

$$1 \leqslant \|T_i\|_2 \leqslant \sqrt{N}\mu$$

- the limit towards low coherence seems well-behaved (all regular properties emerge)
- HOWEVER in average: $$\frac{1}{N} \sum_{i=1}^{N} \|T_i\|_2^2 = 1$$



(a) (b) (c)

(d) (e) (f)

Kernel Localization

The operator T should be understood as kernel localization:

From a kernel $\hat{g}(s)$ generate localized instances:

Kernel Localization

$$\hat{g} : \mathbb{R}^+ \mapsto \mathbb{R} \qquad\qquad T_j g(i) = \sum_\ell \hat{g}(\lambda_\ell) u_\ell(i) u_\ell(j)$$

By functional calculus, the linear operator

$$f \mapsto g(\mathcal{L}) f$$

is the kernelized convolution.

Polynomial Localization

Given a spectral kernel g, construct the family of features:

$$\phi_n(m) = (T_n g)(m) \qquad \phi_n(m) = \sqrt{N} \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell(m) u_\ell^*(n)$$

Are these features localized ?

Polynomial Kernels are K-Localized

$$\widehat{p_K}(\lambda_\ell) = \sum_{k=0}^{K} a_k \lambda_\ell^k \qquad \text{if } d(i,n) > K, \text{ then } (T_i p_K)(n) = 0$$

Polynomial Localization

Given a spectral kernel g, construct the family of features:

$$\phi_n(m) = (T_n g)(m) \qquad \phi_n(m) = \sqrt{N} \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell(m) u_\ell^*(n)$$

Are these features localized ?

Suppose the GFT of the kernel is smooth enough (K+1 different.)

Construct an order K polynomial approximation:

$$\phi'_n(m) = \langle \delta_m, P_K(\mathcal{L}) \delta_n \rangle$$

Exactly localized in a K-ball around n

$$\phi_n(m) = \langle \delta_m, g(\mathcal{L}) \delta_n \rangle$$

Should be well localized within K-ball around n !

Polynomial Localization - Extended

f is $(K{+}1)$-times differentiable:

$$\inf_{q_K}\{\|f - q_K\|_\infty\} \leq \frac{\left[\frac{b-a}{2}\right]^{K+1}}{(K+1)!\;2^K}\|f^{(K+1)}\|_\infty$$

Let $K_{in} := d(i,n) - 1$

$$|(T_i g)(n)| \leq \sqrt{N}\inf_{\widehat{p_{K_{in}}}}\left\{\sup_{\lambda\in[0,\lambda_{\max}]}|\hat{g}(\lambda) - \widehat{p_{K_{in}}}(\lambda)|\right\} = \sqrt{N}\inf_{\widehat{p_{K_{in}}}}\{\|\hat{g} - \widehat{p_{K_{in}}}\|_\infty\}$$

Regular Kernels are Localized

If the kernel is $d(i,n)$-times differentiable:

$$|(T_i g)(n)| \leq \left[\frac{2\sqrt{N}}{d_{in}!}\left(\frac{\lambda_{\max}}{4}\right)^{d_{in}}\sup_{\lambda\in[0,\lambda_{\max}]}|\hat{g}^{(d_{in})}(\lambda)|\right]$$

Polynomial Localization - Extended

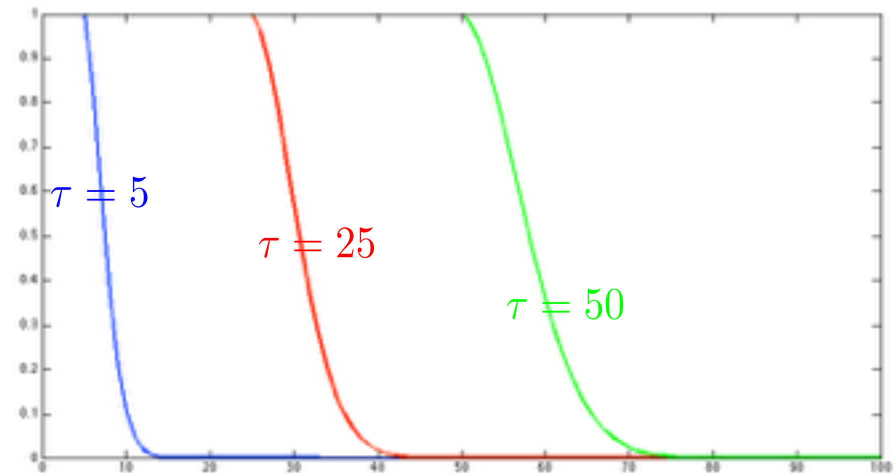
Example: for the heat kernel $\hat{g}(\lambda) = e^{-\tau\lambda}$

$$\frac{|(T_i g)(n)|}{\|T_i g\|_2} \leq \frac{2\sqrt{N}}{d_{in}!}\left(\frac{\tau\lambda_{\max}}{4}\right)^{d_{in}} \leq \sqrt{\frac{2N}{d_{in}\pi}}\, e^{-\frac{1}{12 d_{in}+1}}\left(\frac{\tau\lambda_{\max} e}{4 d_{in}}\right)^{d_{in}}$$

We can estimate an explicit measure of spread in terms of the degrees:

$$\Delta_i^2(f) = \frac{1}{\|f\|_2^2}\sum_{n=1}^{N} d_{in}^2 [f(n)]^2$$

$$\Delta_i^2(T_i g) \leq \frac{\tau N \lambda_{\max} e D_i}{(2\pi)^{\frac{3}{2}}} e^{\frac{\tau\lambda_{\max} e^2 (D_{\max}-1)}{4}}$$



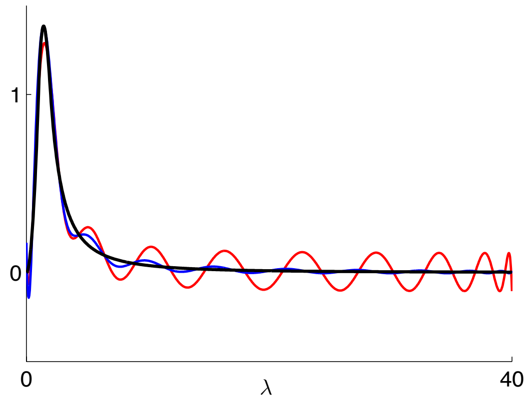
$$\tau \to 0 \Rightarrow T_i g \to \delta_i,\ \Delta_i^2(T_i g) \to 0$$

$$\tau \to +\infty \Rightarrow T_i g \to \frac{1}{\sqrt{N}},\ \Delta_i^2(T_i g) \to \frac{1}{N}\sum_{n=1}^{N} d(i,n)^2$$

Remark on Implementation

Not necessary to compute spectral decomposition

Polynomial approximation : $$\hat{g}(tx) \simeq \sum_{k=0}^{K-1} a_k(t) p_k(x)$$



ex: Chebyshev, minimax

Then wavelet operator expressed with powers of Laplacian:

$$g(t\mathcal{L}) \simeq \sum_{k=0}^{K-1} a_k(t) \mathcal{L}^k$$

And use sparsity of Laplacian in an iterative way

Remark on Implementation

$$\tilde{W}_f(t,j) = \left(p(\mathcal{L})f^{\#}\right)_j \qquad |W_f(t,j) - \tilde{W}_f(t,j)| \leq B\|f\|$$

sup norm control (minimax or Chebyshev)

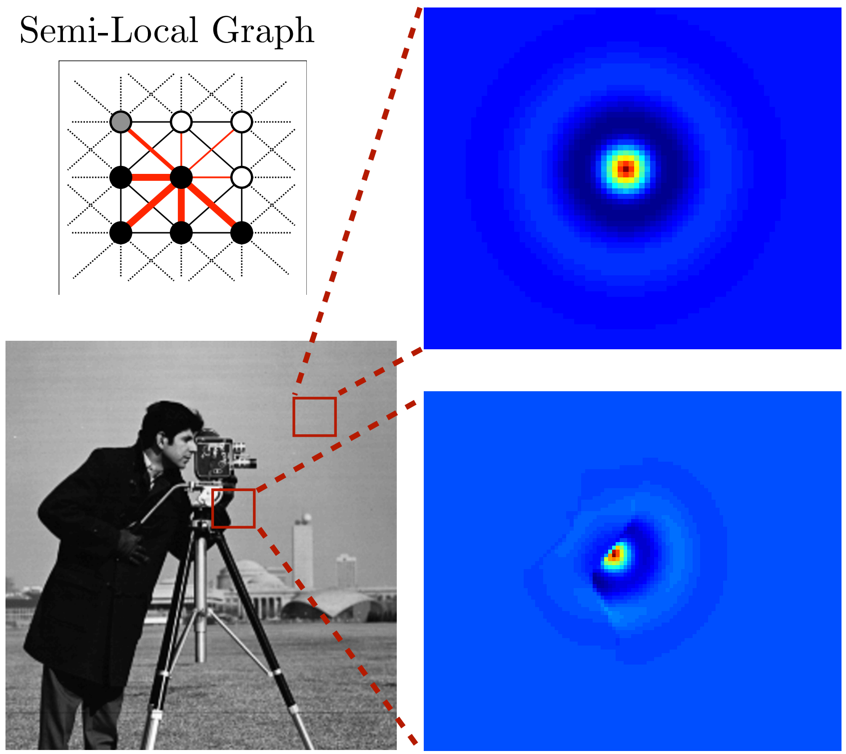
$$\tilde{W}_f(t_n,j) = \left(\frac{1}{2}c_{n,0}f^{\#} + \sum_{k=1}^{M_n} c_{n,k}\overline{T}_k(\mathcal{L})f^{\#}\right)_j$$

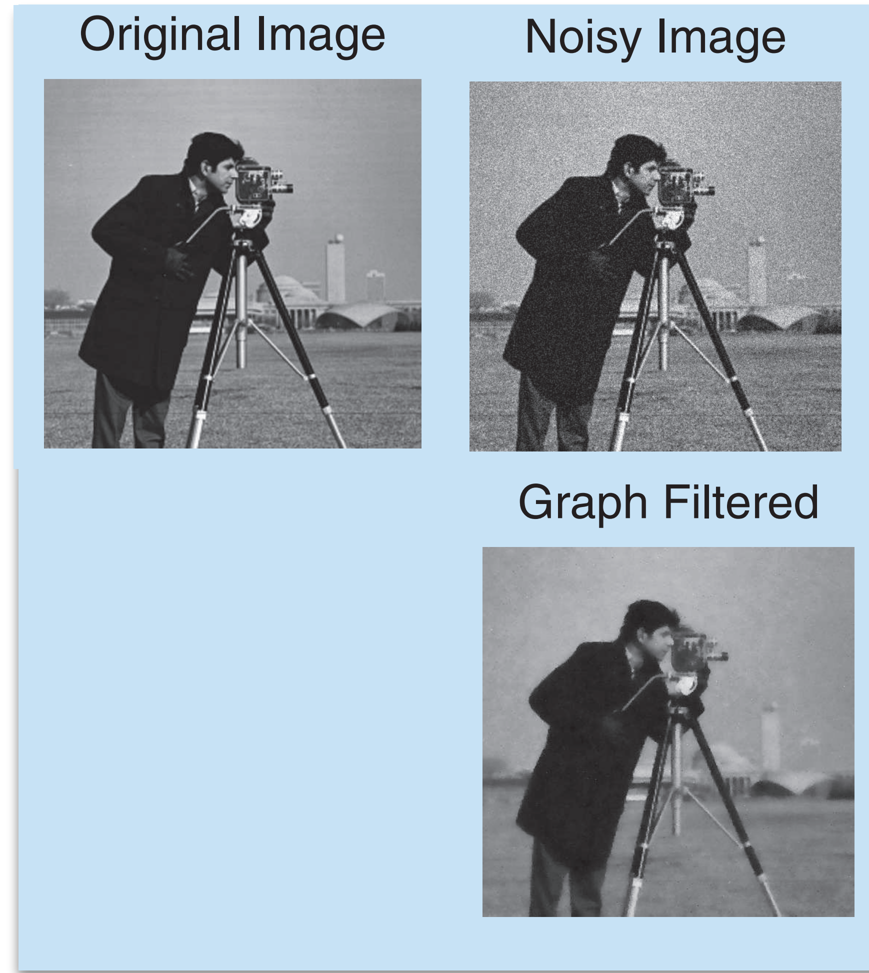
$$\overline{T}_k(\mathcal{L})f = \frac{2}{a_1}(\mathcal{L} - a_2 I)\left(\overline{T}_{k-1}(\mathcal{L})f\right) - \overline{T}_{k-2}(\mathcal{L})f$$

Shifted Chebyshev polynomial

Computational cost dominated by matrix-vector multiply with (sparse) Laplacian matrix

Complexity: $O(\sum_{n=1}^{J} M_n|E|)$ Note: "same" algorithm for adjoint !

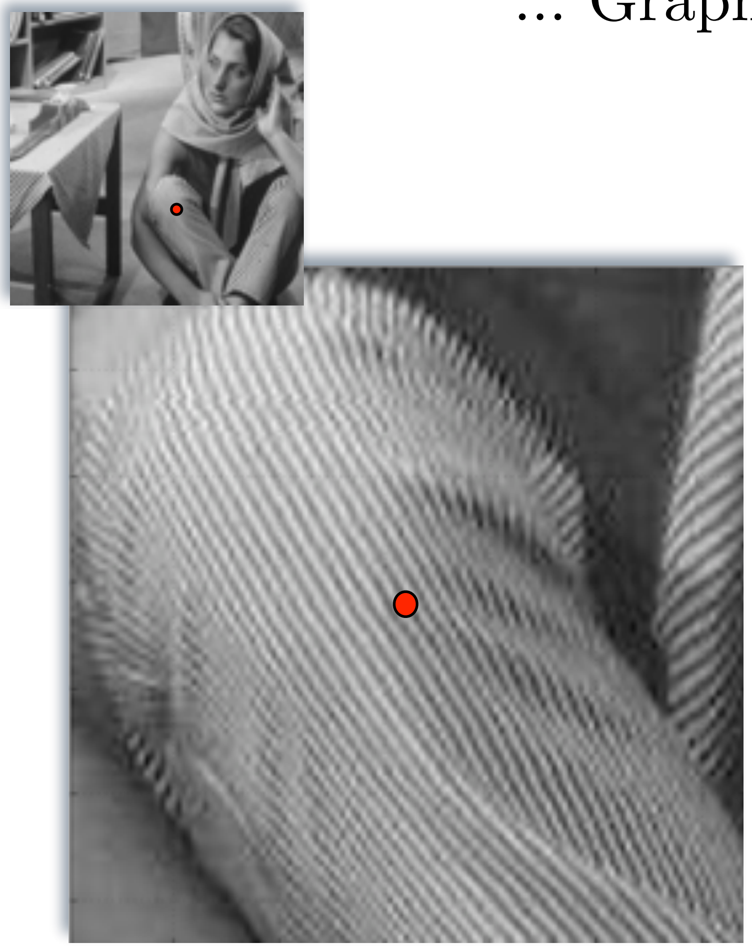

Semi-Local Graph

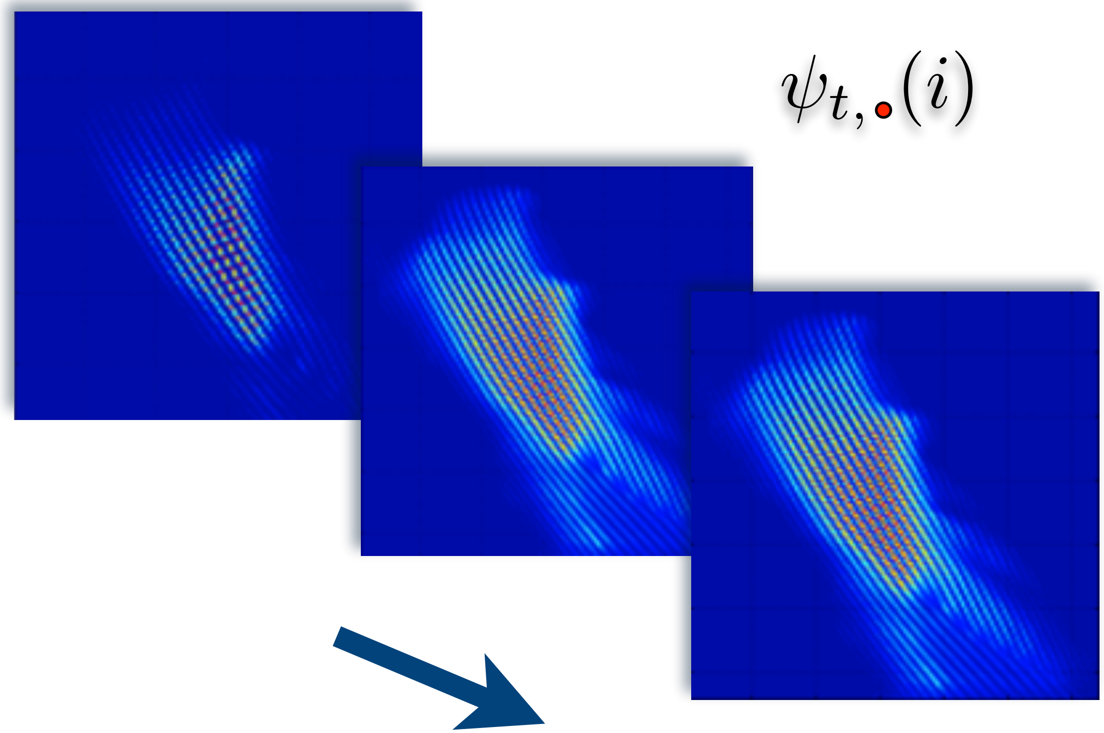

Original Image
Noisy Image
Graph Filtered

Non-local Wavelet Frame

- Non-local Wavelets are ...

... Graph Wavelets on Non-Local Graph





$\psi_{t,\bullet}(i)$

increasing scale

Interest: good *adaptive* sparsity basis

Localization / Uncertainty

Competition between smoothness and localization in the spectral representation of kernels

Remark: $$\sigma_t^2 \sigma_\omega^2 = C \int_{\mathbb{R}} dt |t f(t)|^2 \int_{\mathbb{R}} dt |f'(t)|^2$$

Smooth kernels can be used to construct controlled localized features

Example: Spectral Graph Wavelets

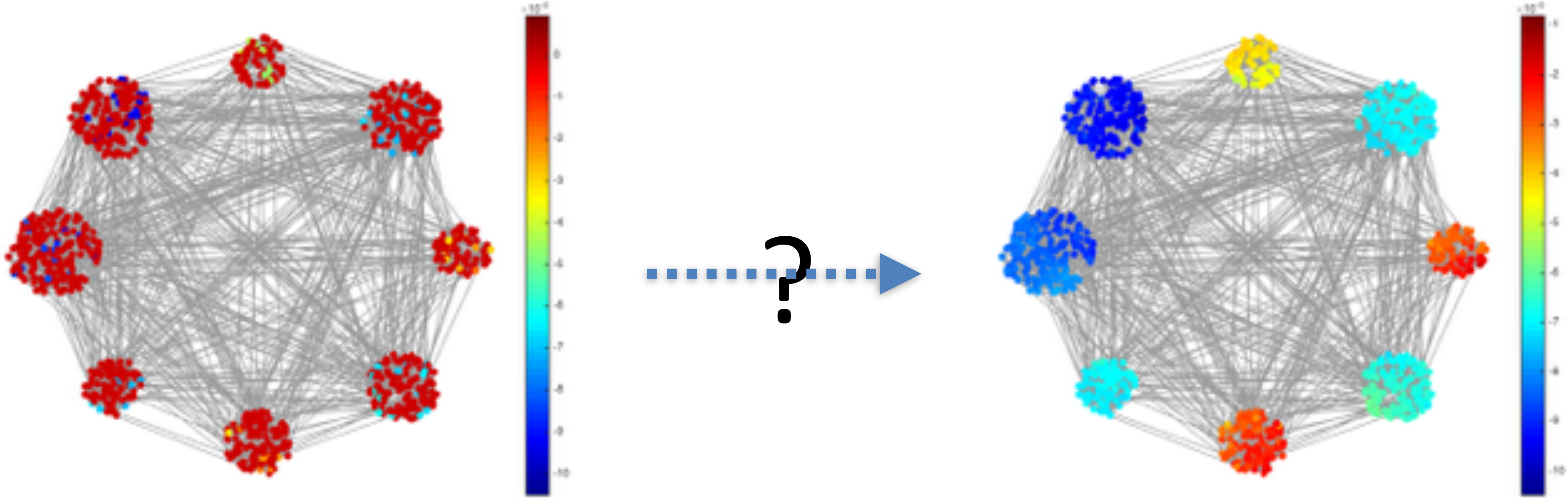
Localization/Smoothness generate sparsity (but more on that later)

Summary so far

- We now have a simple black box theory to design and apply linear filters on graph data
  - results on localisation, uncertainty
  - fast, scalable algorithm
  - all sorts of filter banks studied and used in litterature
- We can use filter banks to construct graph equivalent of linear transforms (wavelets, Gabor,..)
- We can extend stationary signal models
- (sub)-sampling theory

Goal

Given partially observed information at the nodes of a graph



Can we robustly and efficiently infer missing information ?

What signal model ?

How many observations ?

Influence of the structure of the graph ?

Notations

L is real, symmetric PSD

orthonormal eigenvectors $\mathsf{U} \in \mathbb{R}^{n \times n}$ Graph Fourier Matrix

non-negative eigenvalues $\boldsymbol{\lambda}_1 \leqslant \boldsymbol{\lambda}_2 \leqslant \ldots, \boldsymbol{\lambda}_n$

$$\mathsf{L} = \mathsf{U}\Lambda\mathsf{U}^\intercal$$

k-bandlimited signals $\boldsymbol{x} \in \mathbb{R}^n$

Fourier coefficients $\hat{\boldsymbol{x}} = \mathsf{U}^\intercal \boldsymbol{x}$

$$\boldsymbol{x} = \mathsf{U}_k \hat{\boldsymbol{x}}^k \qquad \hat{\boldsymbol{x}}^k \in \mathbb{R}^k$$

$\mathsf{U}_k := (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k) \in \mathbb{R}^{n \times k}$ first k eigenvectors only

Sampling Model

$$\boldsymbol{p} \in \mathbb{R}^n \qquad \boldsymbol{p}_i > 0 \qquad \|\boldsymbol{p}\|_1 = \sum_{i=1}^{n} \boldsymbol{p}_i = 1$$

$$\mathsf{P} := \operatorname{diag}(\boldsymbol{p}) \in \mathbb{R}^{n \times n}$$

Draw independently m samples (random sampling)

$$\mathbb{P}(\omega_j = i) = \boldsymbol{p}_i, \quad \forall j \in \{1, \ldots, m\} \text{ and } \forall i \in \{1, \ldots, n\}$$

$$\boldsymbol{y}_j := \boldsymbol{x}_{\omega_j}, \quad \forall j \in \{1, \ldots, m\}$$

$$\boldsymbol{y} = \mathsf{M}\boldsymbol{x}$$

Sampling Model

$$\frac{\|\mathsf{U}_k^\intercal \boldsymbol{\delta}_i\|_2}{\|\mathsf{U}^\intercal \boldsymbol{\delta}_i\|_2} = \frac{\|\mathsf{U}_k^\intercal \boldsymbol{\delta}_i\|_2}{\|\boldsymbol{\delta}_i\|_2} = \|\mathsf{U}_k^\intercal \boldsymbol{\delta}_i\|_2$$

How much a perfect impulse can be concentrated on first k eigenvectors

Carries interesting information about the graph

Ideally: $\boldsymbol{p}_i$ large wherever $\|\mathsf{U}_k^\intercal \boldsymbol{\delta}_i\|_2$ is large

Graph Coherence

$$\nu_{\boldsymbol{p}}^k := \max_{1 \leqslant i \leqslant n} \left\{ \boldsymbol{p}_i^{-1/2} \|\mathsf{U}_k^\intercal \boldsymbol{\delta}_i\|_2 \right\}$$

Rem: $\nu_{\boldsymbol{p}}^k \geqslant \sqrt{k}$

Stable Embedding

Theorem 1 (Restricted isometry property). *Let* M *be a random subsampling matrix with the sampling distribution* $\boldsymbol{p}$. *For any* $\delta, \epsilon \in (0,1)$, *with probability at least* $1-\epsilon$,

$$(1-\delta)\,\|\boldsymbol{x}\|_2^2 \leqslant \frac{1}{m}\left\|\mathsf{M}\mathsf{P}^{-1/2}\,\boldsymbol{x}\right\|_2^2 \leqslant (1+\delta)\,\|\boldsymbol{x}\|_2^2 \qquad (1)$$

for all $\boldsymbol{x} \in \operatorname{span}(\mathsf{U}_k)$ *provided that*

$$m \geqslant \frac{3}{\delta^2}\,(\nu_{\boldsymbol{p}}^k)^2\,\log\left(\frac{2k}{\epsilon}\right). \qquad (2)$$

$\mathsf{M}\mathsf{P}^{-1/2}\,\boldsymbol{x} = \mathsf{P}_\Omega^{-1/2}\mathsf{M}\boldsymbol{x}$ Only need M, re-weighting offline

$(\nu_{\boldsymbol{p}}^k)^2 \geqslant k$ Need to sample at least k nodes

Proof similar to CS in bounded ONB but simpler since model is a subspace (not a union)

Stable Embedding

$$(\nu_{\boldsymbol{p}}^k)^2 \geqslant k \qquad \text{Need to sample at least } k \text{ nodes}$$

Can we reduce to optimal amount ?

Variable Density Sampling $$\boldsymbol{p}_i^* := \frac{\|\mathsf{U}_k^\intercal \boldsymbol{\delta}_i\|_2^2}{k}, \quad i = 1, \dots, n$$

is such that: $(\nu_{\boldsymbol{p}}^k)^2 = k$ and depends on structure of graph

Corollary 1. *Let* M *be a random subsampling matrix constructed with the sampling distribution* $\boldsymbol{p}^*$. *For any* $\delta, \epsilon \in (0,1)$, *with probability at least* $1 - \epsilon$,

$$(1-\delta)\,\|\boldsymbol{x}\|_2^2 \leqslant \frac{1}{m}\left\|\mathsf{M}\mathsf{P}^{-1/2}\,\boldsymbol{x}\right\|_2^2 \leqslant (1+\delta)\,\|\boldsymbol{x}\|_2^2$$

for all $\boldsymbol{x} \in \mathrm{span}(\mathsf{U}_k)$ *provided that*

$$m \geqslant \frac{3}{\delta^2}\, k\, \log\left(\frac{2k}{\epsilon}\right).$$

Recovery Procedures

$$\boldsymbol{y} = \mathsf{M}\boldsymbol{x} + \boldsymbol{n} \qquad \boldsymbol{y} \in \mathbb{R}^m$$

$$\boldsymbol{x} \in \operatorname{span}(\mathsf{U}_k)$$ stable embedding

Standard Decoder

$$\min_{\boldsymbol{z} \in \operatorname{span}(\mathsf{U}_k)} \left\| \mathsf{P}_\Omega^{-1/2} (\mathsf{M}\boldsymbol{z} - \boldsymbol{y}) \right\|_2$$

need projector

re-weighting for RIP

Recovery Procedures

$$\boldsymbol{y} = \mathsf{M}\boldsymbol{x} + \boldsymbol{n} \qquad \boldsymbol{y} \in \mathbb{R}^m$$

$\boldsymbol{x} \in \mathrm{span}(\mathsf{U}_k)$ stable embedding

Efficient Decoder:

$$\min_{\boldsymbol{z} \in \mathbb{R}^n} \left\| \mathsf{P}_\Omega^{-1/2} (\mathsf{M}\boldsymbol{z} - \boldsymbol{y}) \right\|_2^2 + \gamma\, \boldsymbol{z}^\intercal g(\mathsf{L}) \boldsymbol{z}$$

soft constrain on frequencies

efficient implementation

Analysis of Standard Decoder

Standard Decoder:

$$\min_{\boldsymbol{z} \in \mathrm{span}(\mathsf{U}_k)} \left\| \mathsf{P}_\Omega^{-1/2} \left(\mathsf{M}\boldsymbol{z} - \boldsymbol{y} \right) \right\|_2$$

Theorem 1. *Let Ω be a set of m indices selected independently from $\{1, \ldots, n\}$ with sampling distribution $\boldsymbol{p} \in \mathbb{R}^n$, and M the associated sampling matrix. Let $\epsilon, \delta \in (0, 1)$ and $m \geqslant \frac{3}{\delta^2} \, (\nu_{\boldsymbol{p}}^k)^2 \, \log\left(\frac{2k}{\epsilon}\right)$. With probability at least $1 - \epsilon$, the following holds for all $\boldsymbol{x} \in \mathrm{span}(\mathsf{U}_k)$ and all $\boldsymbol{n} \in \mathbb{R}^m$.*

i) Let $\boldsymbol{x}^*$ be the solution of Standard Decoder with $\boldsymbol{y} = \mathsf{M}\boldsymbol{x} + \boldsymbol{n}$. Then,

$$\|\boldsymbol{x}^* - \boldsymbol{x}\|_2 \leqslant \frac{2}{\sqrt{m\,(1-\delta)}} \left\| \mathsf{P}_\Omega^{-1/2} \boldsymbol{n} \right\|_2 . \qquad (1)$$

Exact recovery when noiseless

ii) There exist particular vectors $\boldsymbol{n}_0 \in \mathbb{R}^m$ such that the solution $\boldsymbol{x}^*$ of Standard Decoder with $\boldsymbol{y} = \mathsf{M}\boldsymbol{x} + \boldsymbol{n}_0$ satisfies

$$\|\boldsymbol{x}^* - \boldsymbol{x}\|_2 \geqslant \frac{1}{\sqrt{m\,(1+\delta)}} \left\| \mathsf{P}_\Omega^{-1/2} \boldsymbol{n}_0 \right\|_2 . \qquad (2)$$

Analysis of Efficient Decoder

Efficient Decoder:

$$\min_{\boldsymbol{z}\in\mathbb{R}^n} \left\| \mathsf{P}_\Omega^{-1/2} \left(\mathsf{M}\boldsymbol{z} - \boldsymbol{y} \right) \right\|_2^2 + \gamma\, \boldsymbol{z}^\intercal g(\mathsf{L})\boldsymbol{z}$$

non-negative

Filter reshapes Fourier coefficients

$$h : \mathbb{R} \to \mathbb{R} \qquad \boldsymbol{x}_h := \mathsf{U}\,\mathrm{diag}(\hat{\boldsymbol{h}})\,\mathsf{U}^\intercal \boldsymbol{x} \in \mathbb{R}^n$$

$$\hat{\boldsymbol{h}} = (h(\boldsymbol{\lambda}_1), \ldots, h(\boldsymbol{\lambda}_n))^\intercal \in \mathbb{R}^n$$

$$p(t) = \sum_{i=0}^{d} \alpha_i\, t^i \qquad \boldsymbol{x}_p = \mathsf{U}\,\mathrm{diag}(\hat{\boldsymbol{p}})\,\mathsf{U}^\intercal \boldsymbol{x} = \sum_{i=0}^{d} \alpha_i\, \mathsf{L}^i \boldsymbol{x}$$

Pick special polynomials and use e.g. recurrence relations for fast filtering
(with sparse matrix-vector multiply only)

Analysis of Efficient Decoder

Efficient Decoder:

$$\min_{\boldsymbol{z}\in\mathbb{R}^n} \left\| \mathsf{P}_\Omega^{-1/2} \left(\mathsf{M}\boldsymbol{z} - \boldsymbol{y} \right) \right\|_2^2 + \gamma \, \boldsymbol{z}^\mathsf{T} g(\mathsf{L}) \boldsymbol{z}$$

non-negative

non-decreasing $\Longrightarrow$
penalizes high-frequencies

Favours reconstruction of approximately band-limited signals

Ideal filter yields Standard Decoder

$$i_{\lambda_k}(t) := \begin{cases} 0 & \text{if } t \in [0, \lambda_k], \\ +\infty & \text{otherwise,} \end{cases}$$

Analysis of Efficient Decoder

Theorem 1. *Let* Ω, M, P, m *as before and* $M_{\max} > 0$ *be a constant such that* $\left\|\mathsf{M}\mathsf{P}^{-1/2}\right\|_2 \leqslant M_{\max}$. *Let* $\epsilon, \delta \in (0,1)$. *With probability at least* $1-\epsilon$, *the following holds for all* $\boldsymbol{x} \in \operatorname{span}(\mathsf{U}_k)$, *all* $\boldsymbol{n} \in \mathbb{R}^n$, *all* $\gamma > 0$, *and all nonnegative and nondecreasing polynomial functions* g *such that* $g(\boldsymbol{\lambda}_{k+1}) > 0$.

Let $\boldsymbol{x}^*$ *be the solution of Efficient Decoder with* $\boldsymbol{y} = \mathsf{M}\boldsymbol{x} + \boldsymbol{n}$. *Then,*

$$\|\boldsymbol{\alpha}^* - \boldsymbol{x}\|_2 \leqslant \frac{1}{\sqrt{m(1-\delta)}} \left[\left(2 + \frac{M_{\max}}{\sqrt{\gamma g(\boldsymbol{\lambda}_{k+1})}} \right) \left\| \mathsf{P}_\Omega^{-1/2} \boldsymbol{n} \right\|_2 + \left(M_{\max} \sqrt{\frac{g(\boldsymbol{\lambda}_k)}{g(\boldsymbol{\lambda}_{k+1})}} + \sqrt{\gamma g(\boldsymbol{\lambda}_k)} \right) \|\boldsymbol{x}\|_2 \right], \tag{1}$$

and

$$\|\boldsymbol{\beta}^*\|_2 \leqslant \frac{1}{\sqrt{\gamma g(\boldsymbol{\lambda}_{k+1})}} \left\| \mathsf{P}_\Omega^{-1/2} \boldsymbol{n} \right\|_2 + \sqrt{\frac{g(\boldsymbol{\lambda}_k)}{g(\boldsymbol{\lambda}_{k+1})}} \|\boldsymbol{x}\|_2, \tag{2}$$

where $\boldsymbol{\alpha}^* := \mathsf{U}_k \mathsf{U}_k^\intercal \boldsymbol{x}^*$ *and* $\boldsymbol{\beta}^* := (\mathsf{I} - \mathsf{U}_k \mathsf{U}_k^\intercal)\, \boldsymbol{x}^*$.

Analysis of Efficient Decoder

Noiseless case:

$$\|\boldsymbol{x}^* - \boldsymbol{x}\|_2 \leqslant \frac{1}{\sqrt{m(1-\delta)}} \left(M_{\max} \sqrt{\frac{g(\boldsymbol{\lambda}_k)}{g(\boldsymbol{\lambda}_{k+1})}} + \sqrt{\gamma g(\boldsymbol{\lambda}_k)} \right) \|\boldsymbol{x}\|_2 + \sqrt{\frac{g(\boldsymbol{\lambda}_k)}{g(\boldsymbol{\lambda}_{k+1})}} \|\boldsymbol{x}\|_2$$

$g(\boldsymbol{\lambda}_k) = 0$ + non-decreasing implies perfect reconstruction

Otherwise:

choose γ as close as possible to 0 and seek to minimise the ratio $g(\boldsymbol{\lambda}_k)/g(\boldsymbol{\lambda}_{k+1})$

Choose filter to increase spectral gap ?

Clusters are of course good

Noise: $\|\mathsf{P}_\Omega^{-1/2}\boldsymbol{n}\|_2 / \|\boldsymbol{x}\|_2$

Estimating the Optimal Distribution

Need to estimate $\|\mathsf{U}_k^\intercal \boldsymbol{\delta}_i\|_2^2$

Filter random signals with ideal low-pass filter:

$$\boldsymbol{r}_{b_{\boldsymbol{\lambda}_k}} = \mathsf{U}\,\mathrm{diag}(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_k, 0, \ldots, 0)\,\mathsf{U}^\intercal\,\boldsymbol{r} = \mathsf{U}_k \mathsf{U}_k^\intercal\,\boldsymbol{r}$$

$$\mathbb{E}\left(\boldsymbol{r}_{b_{\boldsymbol{\lambda}_k}}\right)_i^2 = \boldsymbol{\delta}_i^\intercal \mathsf{U}_k \mathsf{U}_k^\intercal\,\mathbb{E}(\boldsymbol{r}\boldsymbol{r}^\intercal)\,\mathsf{U}_k \mathsf{U}_k^\intercal \boldsymbol{\delta}_i = \|\mathsf{U}_k^\intercal \boldsymbol{\delta}_i\|_2^2$$

In practice, one may use a polynomial approximation of the ideal filter and:

$$\tilde{\boldsymbol{p}}_i := \frac{\sum_{l=1}^{L} \left(\boldsymbol{r}^l_{c_{\lambda_k}}\right)_i^2}{\sum_{i=1}^{n} \sum_{l=1}^{L} \left(\boldsymbol{r}^l_{c_{\lambda_k}}\right)_i^2}$$

$$L \geqslant \frac{C}{\delta^2} \log\left(\frac{2n}{\epsilon}\right)$$

Estimating the Eigengap

Again, low-pass filtering random signals:

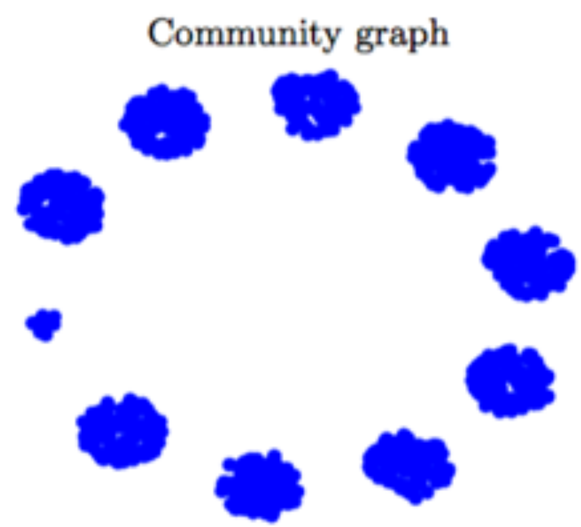
$$(1-\delta)\sum_{i=1}^{n}\left\|\mathsf{U}_{j^*}^{\mathsf{T}}\boldsymbol{\delta}_i\right\|_2^2 \leqslant \sum_{i=1}^{n}\sum_{l=1}^{L}(\boldsymbol{r}_{b_\lambda}^l)_i^2 \leqslant (1+\delta)\sum_{i=1}^{n}\left\|\mathsf{U}_{j^*}^{\mathsf{T}}\boldsymbol{\delta}_i\right\|_2^2$$

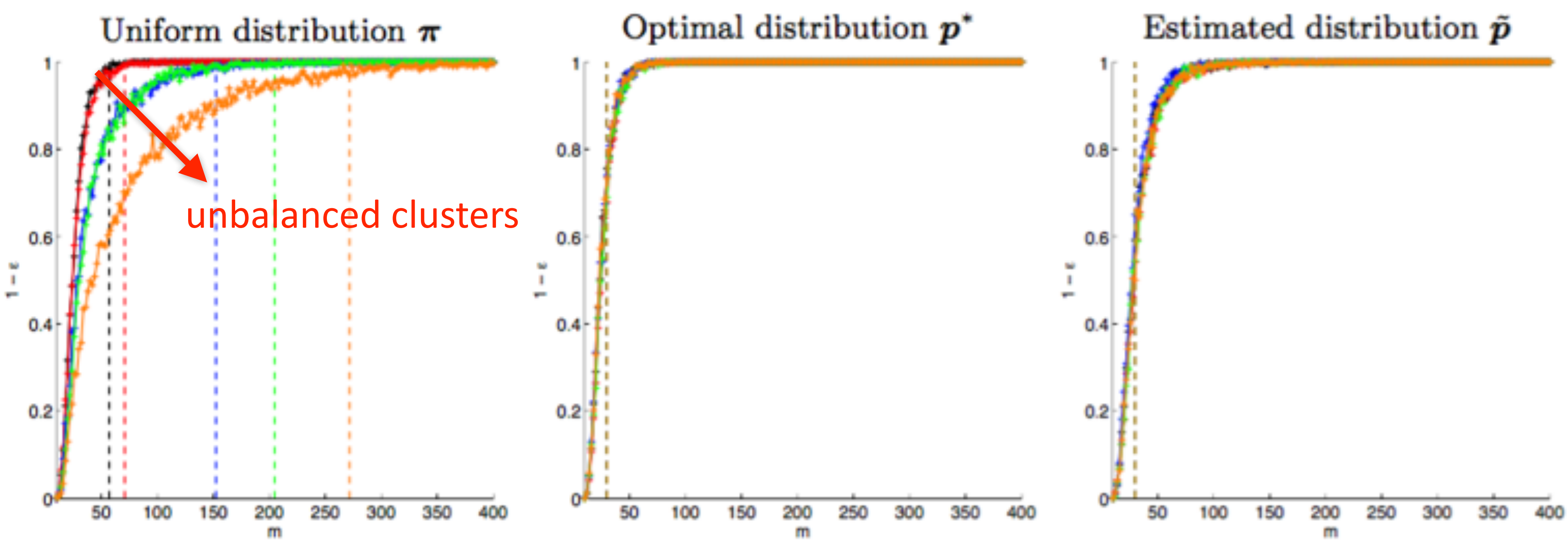
Since: $\sum_{i=1}^{n}\left\|\mathsf{U}_{j^*}^{\mathsf{T}}\boldsymbol{\delta}_i\right\|_2^2 = \|\mathsf{U}_{j^*}\|_{\text{Frob}}^2 = j^*$

We have: $(1-\delta)\, j^* \leqslant \sum_{i=1}^{n}\sum_{l=1}^{L}(\boldsymbol{r}_{b_\lambda}^l)_i^2 \leqslant (1+\delta)\, j^*$

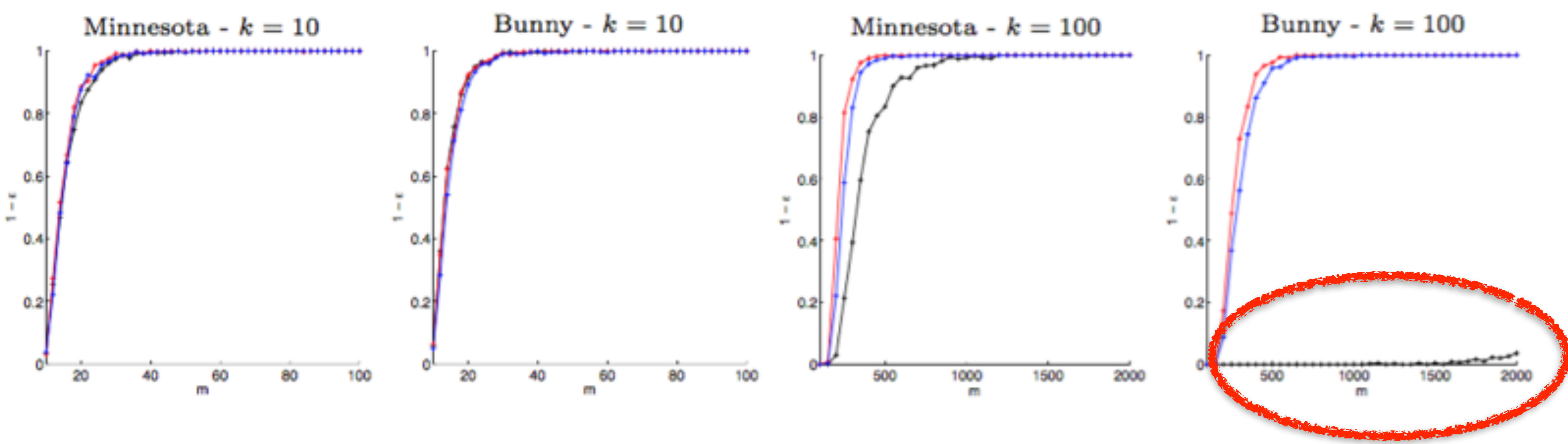
Dichotomy using the filter bandwidth

Experiments

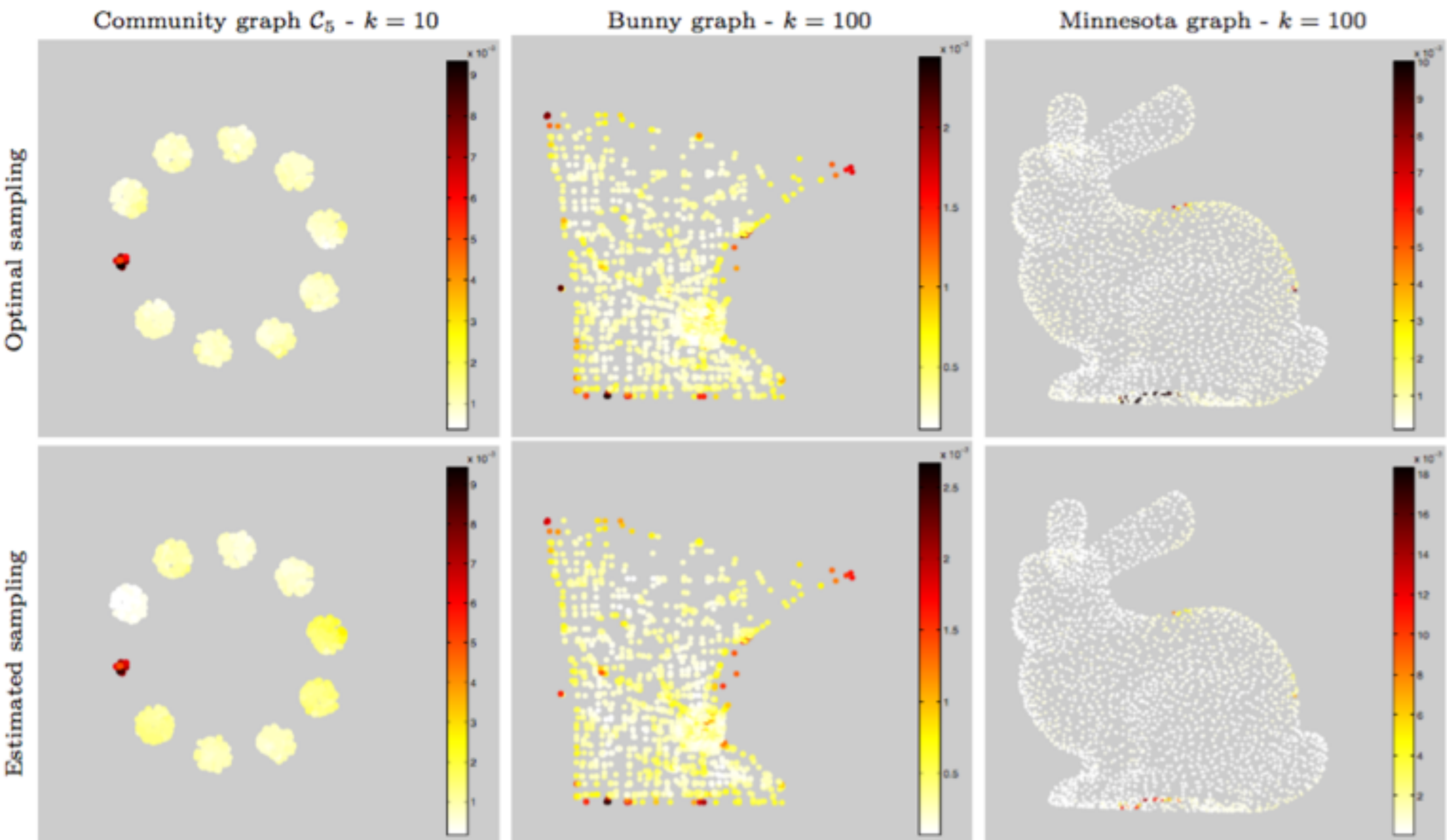




Experiments



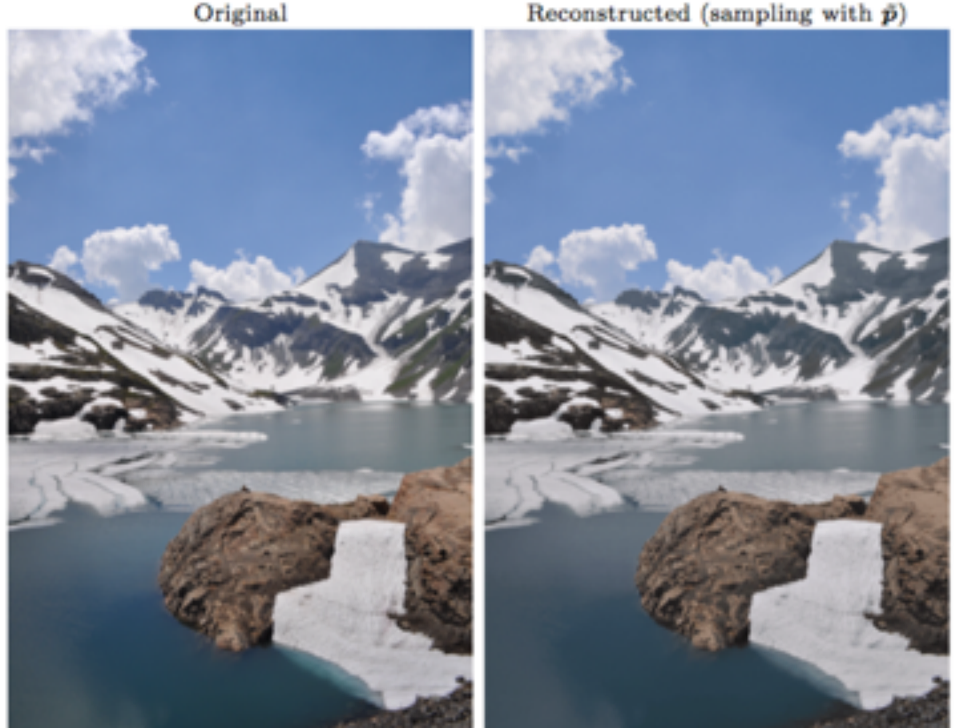
Experiments



Experiments

(a) (b)

Original — Reconstructed (sampling with $\tilde{\boldsymbol{p}}$)



7%

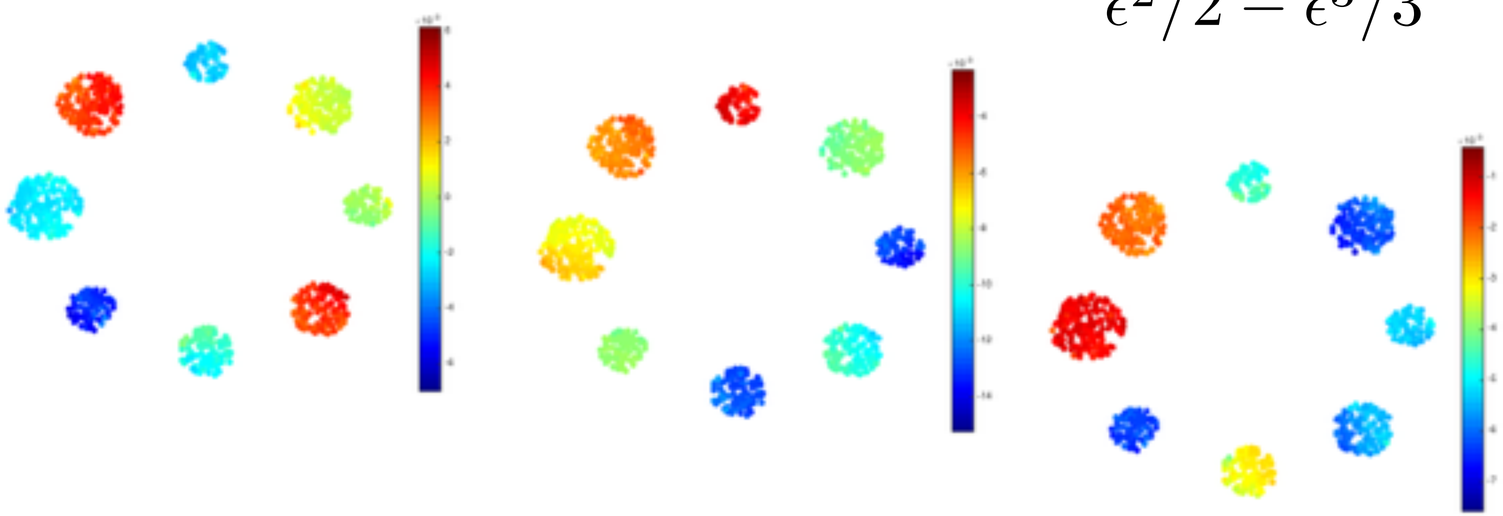
Compressive Spectral Clustering

Clustering equivalent to recovery of cluster assignment functions

Well-defined clusters -> band-limited assignment functions!

Generate features by filtering random signals

by Johnson-Lindenstrauss $\eta = \dfrac{4 + 2\beta}{\epsilon^2/2 - \epsilon^3/3} \log n$



Compressive Spectral Clustering

Clustering equivalent to recovery of cluster assignment functions

Well-defined clusters -> band-limited assignment functions!

Generate features by filtering random signals

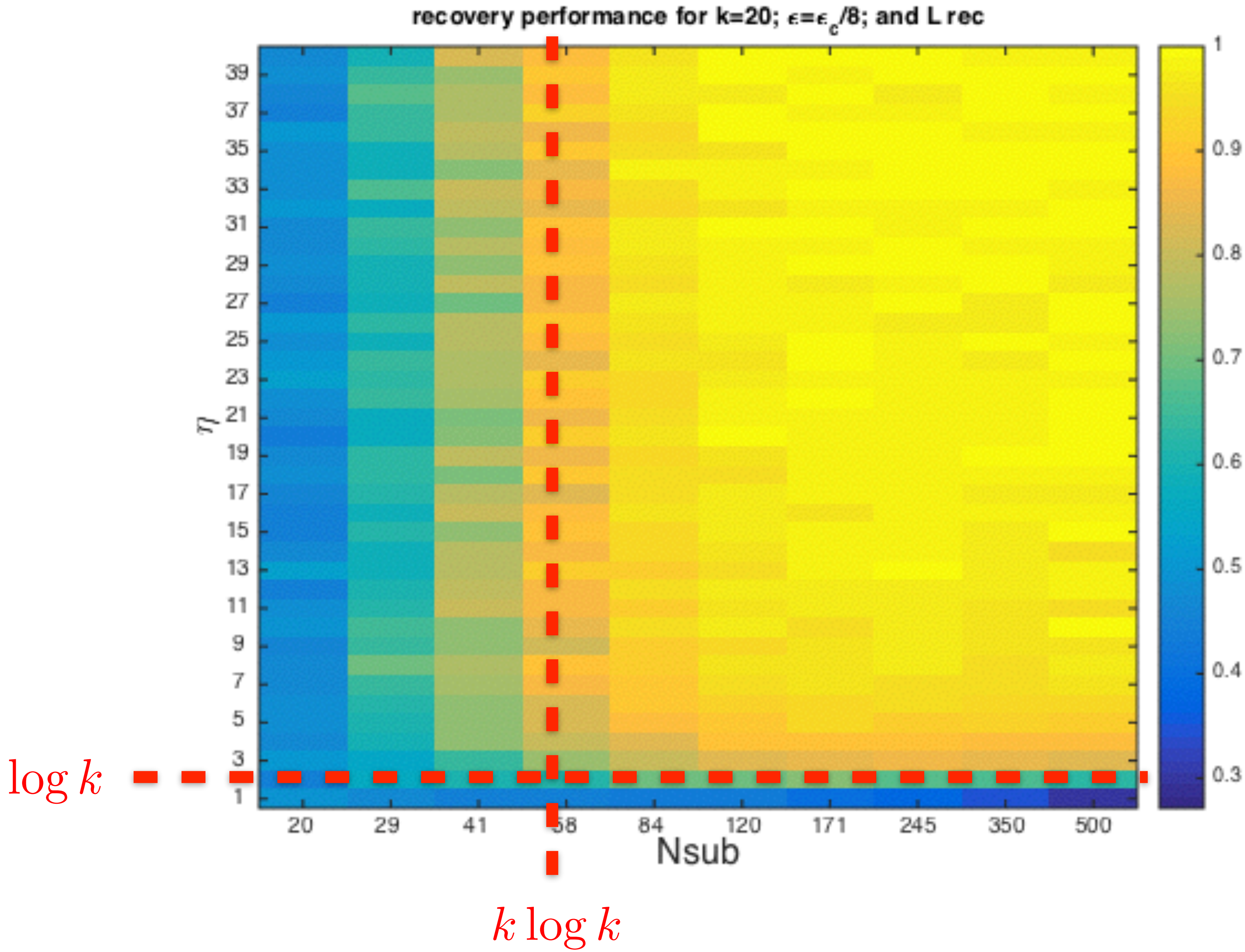
by Johnson-Lindenstrauss $\eta = \frac{4 + 2\beta}{\epsilon^2/2 - \epsilon^3/3} \log n$

Each feature map is smooth, therefore keep

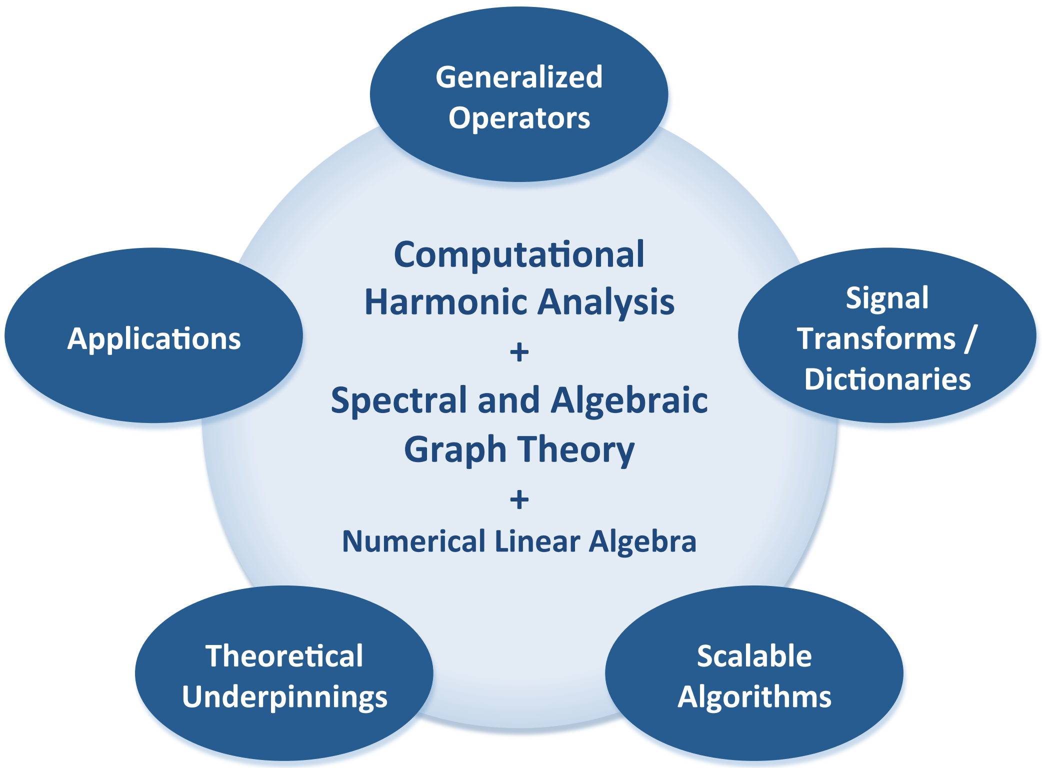
$$m \geqslant \frac{6}{\delta^2} \nu_k^2 \log\left(\frac{k}{\epsilon'}\right)$$

Use k-means on compressed data and feed into Efficient Decoder

Compressive Spectral Clustering



Outlook



- Application of graph signal processing techniques to real science and engineering problems is in its infancy
- Connections with "traditional" signal processing, machine learning, ...

Thank you !