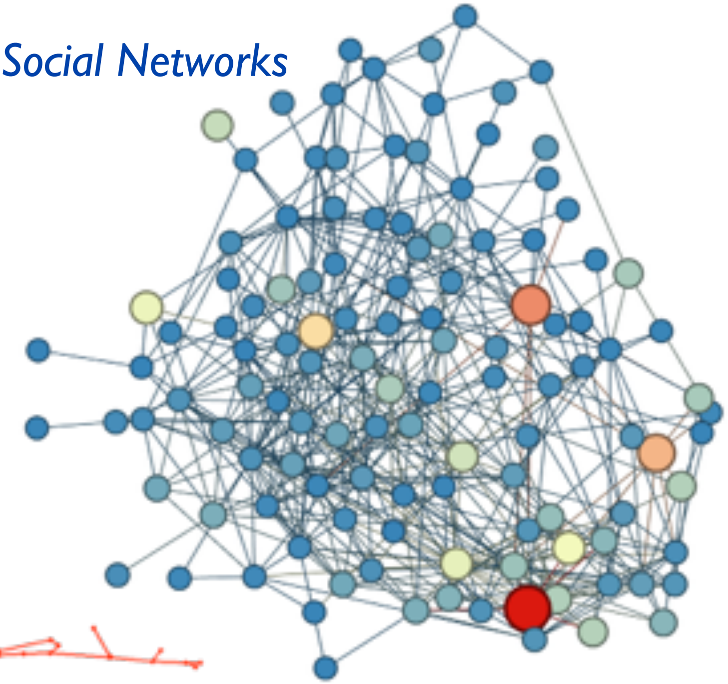
Harmonic Analysis on Graphs and Networks

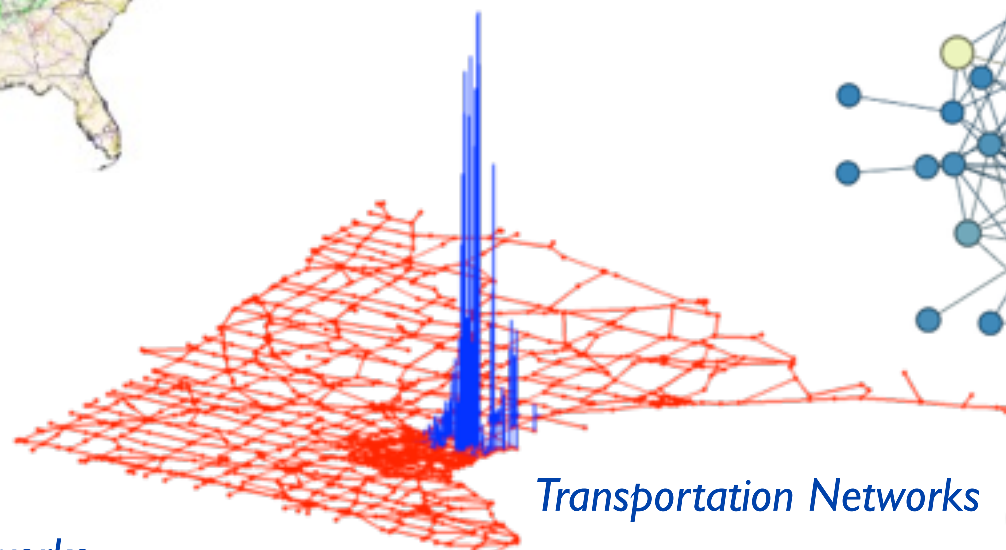
Pierre Vandergheynst
Signal Processing Lab
Swiss Federal Institute of Technology

Signal Processing on Graphs

Energy Networks

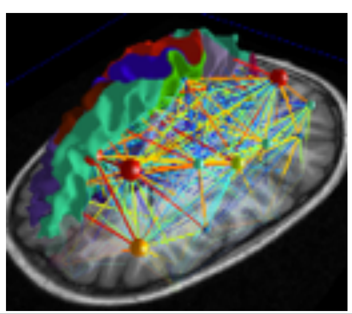
Social Networks

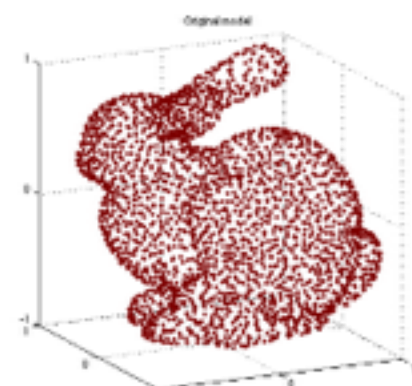




Transportation Networks

Biological Networks



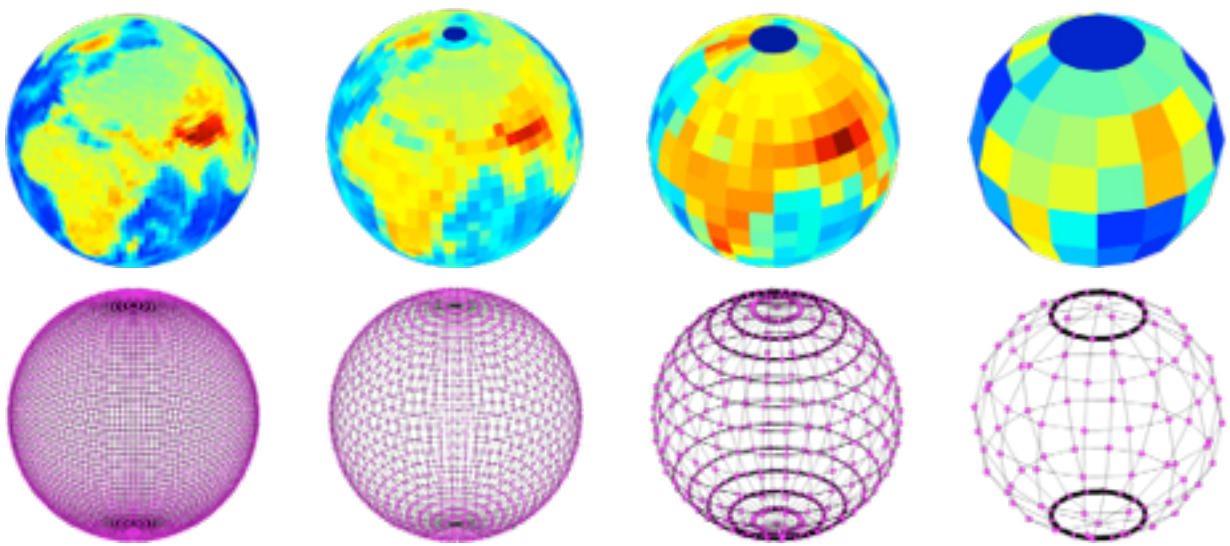


Irregular Data Domains


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

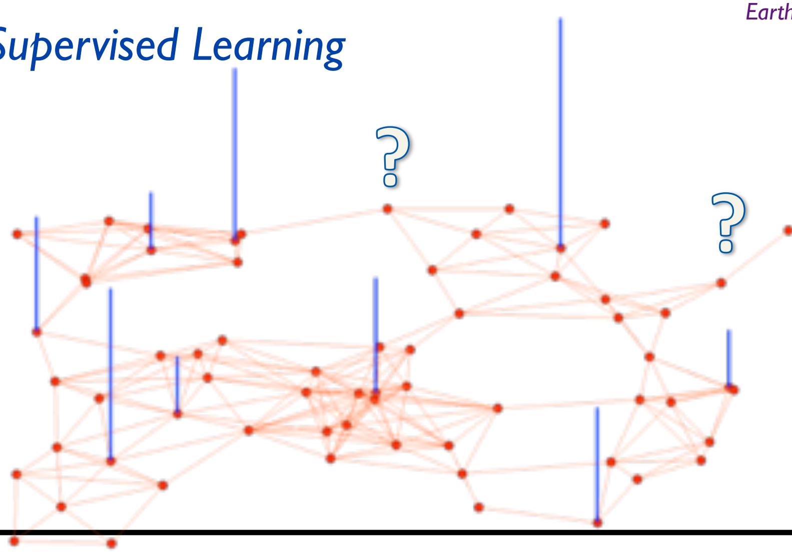
Compression / Visualization

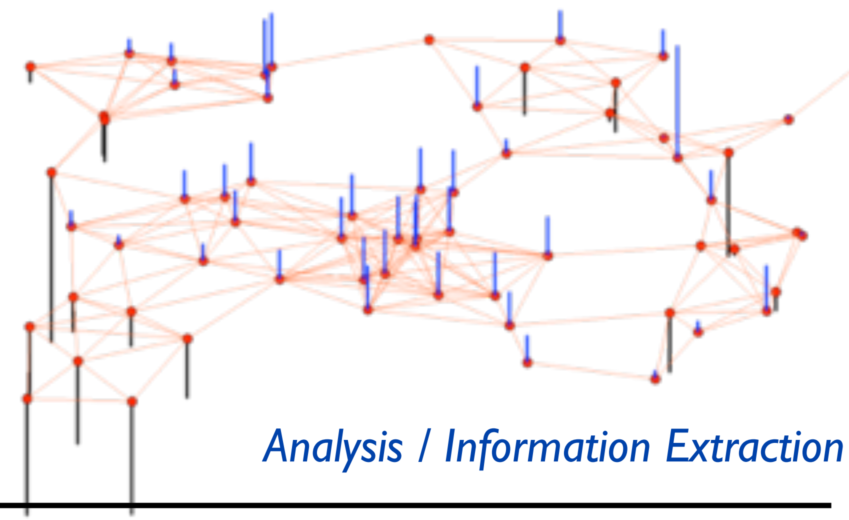


Earth data source: Frederik Simons

Denoising

Semi-Supervised Learning

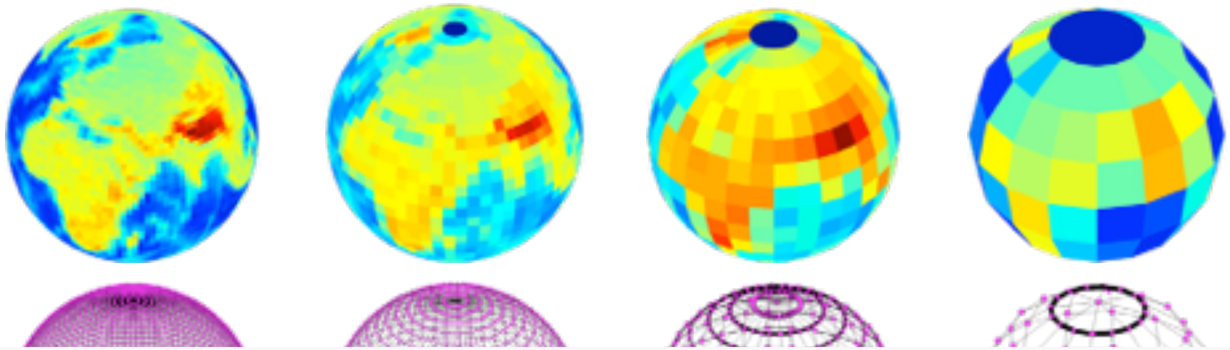


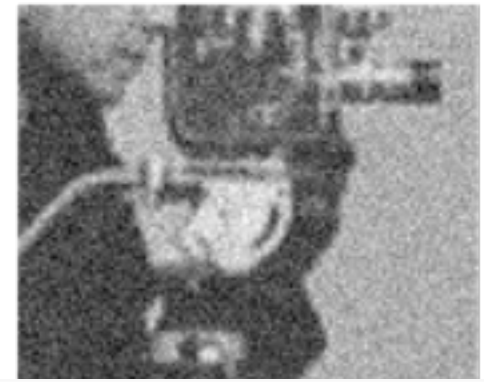


Analysis / Information Extraction

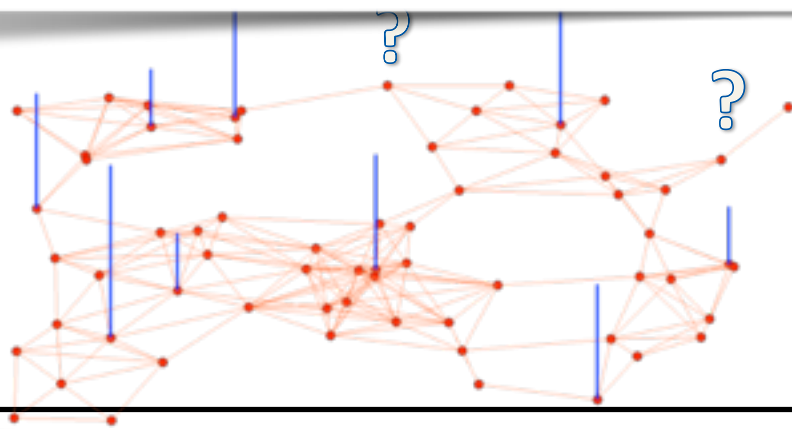
Some Typical Processing Problems

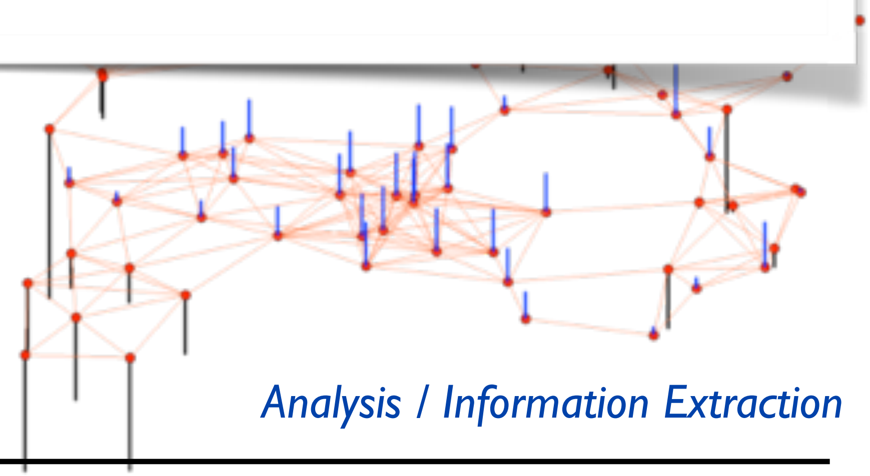
Compression / Visualization





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
- Kernel Convolution:
  - Localization, filtering, smoothing and applications
- Spectral Graph Wavelets
- Multiresolution
- From Graphs to Manifolds

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 $\mathbf{D} = \mathrm{diag}(\mathrm{d})$ the degree matrix

Graph Laplacians, Signals on Graphs

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

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

Laplacian as an operator on space of graph signals

$$\mathcal{L}f(u) = \sum_{v \sim u} \left(f(u) - f(v)\right)$$

Some differential operators

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

Explicit forms:

$$\mathbf{S} = \begin{array}{cc} \begin{array}{c} e=(u,v) \\ \begin{pmatrix} & & \\ \cdots & -1 & \cdots \\ & & \\ \cdots & 1 & \cdots \\ & & \end{pmatrix} \end{array} & \begin{array}{c} \\ \\ u \\ \\ v \\ \\ \end{array} \end{array}$$

$$\mathbf{S}^* f(u,v) = f(v) - f(u)$$ 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: $\langle f, \mathcal{L} f \rangle = \sum_{u \sim v} \big(f(u) - f(v)\big)^2 \geq 0$ 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 $\|\nabla_u f\|_2 = \sqrt{\sum_{v \sim u} |S^* f(u,v)|^2}$

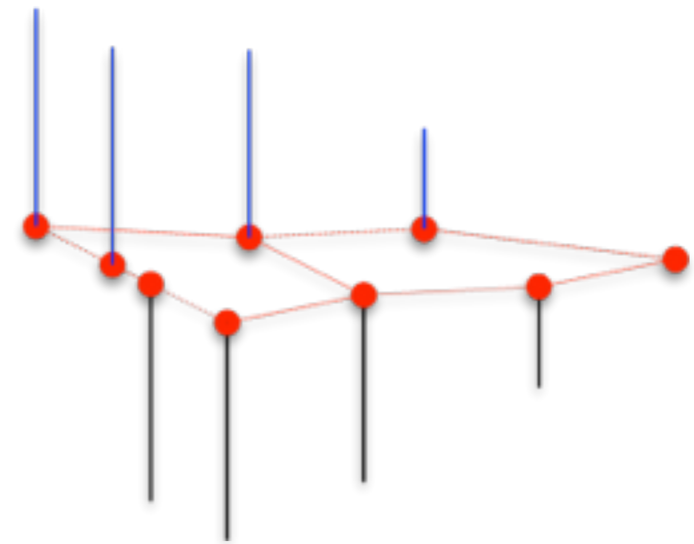
Total variation $|f|_{TV} = \sum_{u \in V} \|\nabla_u f\|_2 = \sum_{u \in V} \sqrt{\sum_{v \sim u} |S^* f(u,v)|^2}$

Notions of Global Regularity for Graph

Discrete Calculus, Grady and Polimeni, 2010

Edge Derivative

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

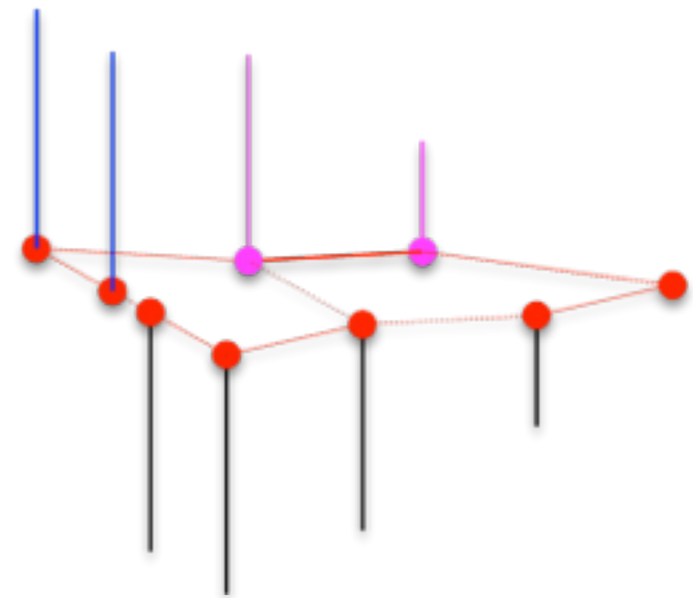


Notions of Global Regularity for Graph

Discrete Calculus, Grady and Polimeni, 2010

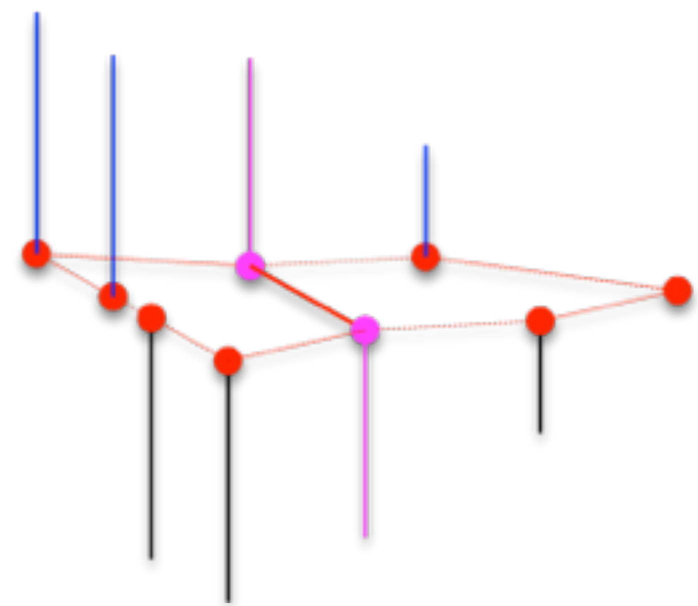
Edge Derivative

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



Notions of Global Regularity for Graph

Discrete Calculus, Grady and Polimeni, 2010



Edge Derivative

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

Notions of Global Regularity for Graph

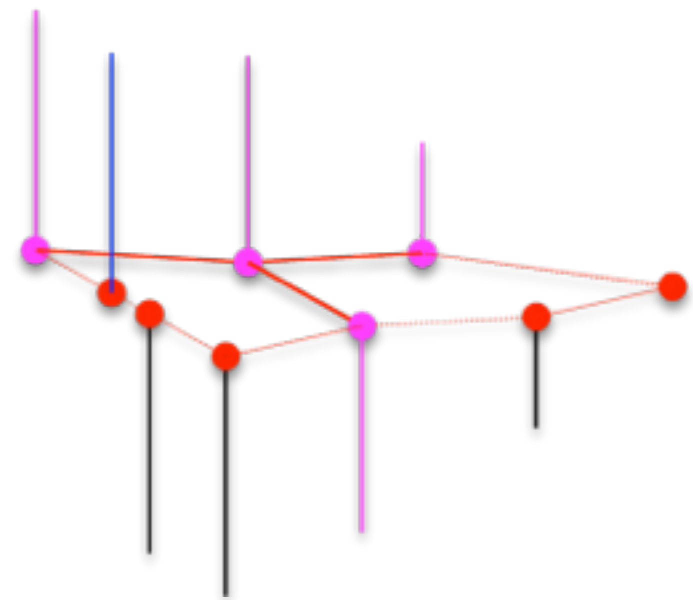
Discrete Calculus, Grady and Polimeni, 2010

Edge Derivative

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

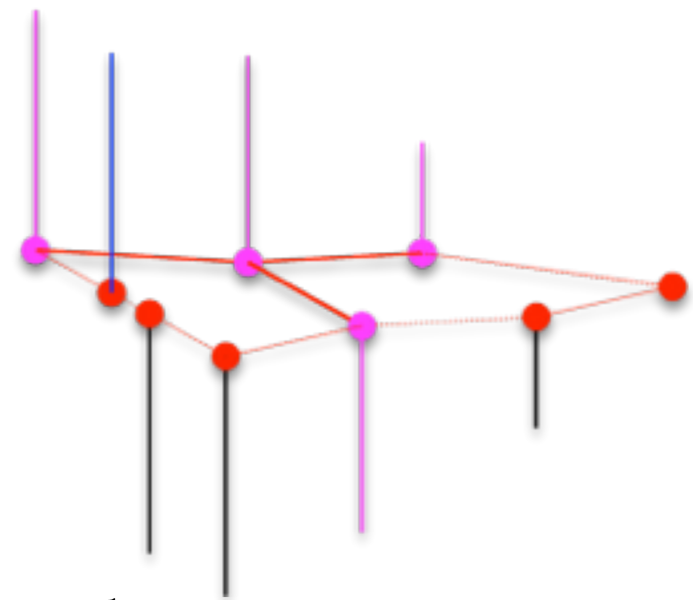
Graph Gradient

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



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

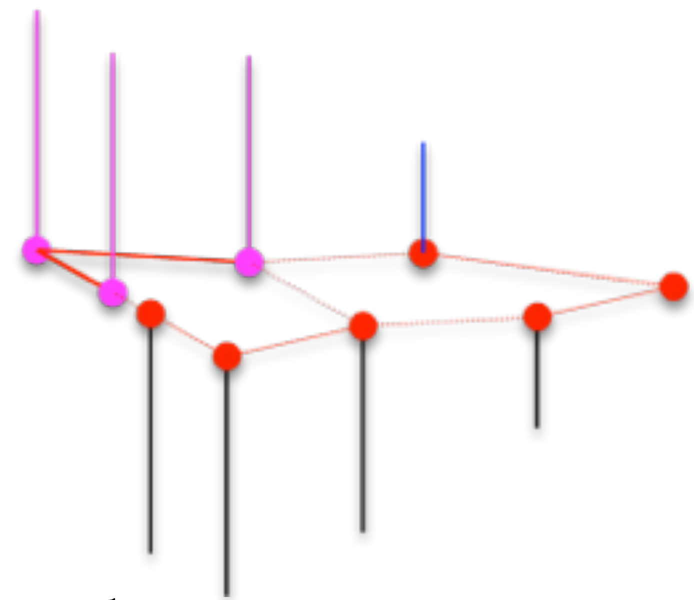
$$\nabla_m \mathbf{f} := \left[\left\{ \left.\frac{\partial \mathbf{f}}{\partial e}\right|_m \right\}_{e \in \mathcal{E} \text{ s.t. } e=(m,n)} \right]$$

Local Variation

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

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

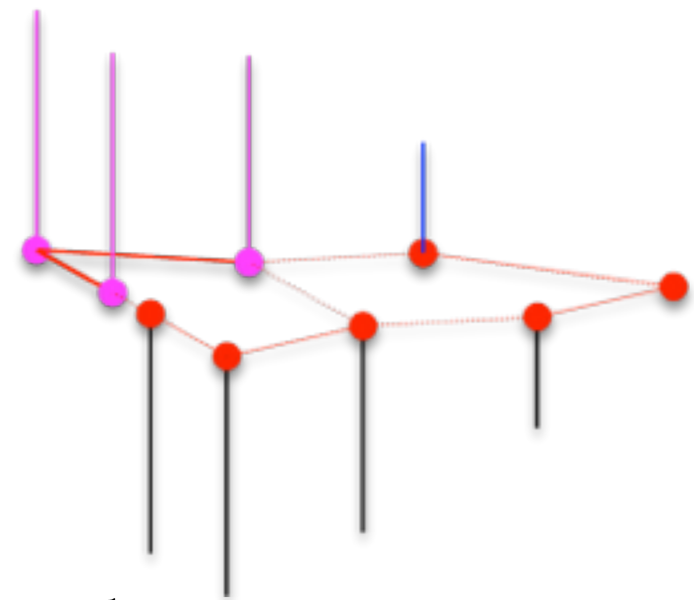
$$\nabla_m \mathbf{f} := \left[\left\{\left.\frac{\partial \mathbf{f}}{\partial e}\right|_m\right\}_{e\in\mathcal{E} \text{ s.t. } e=(m,n)}\right]$$

Local Variation

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

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

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

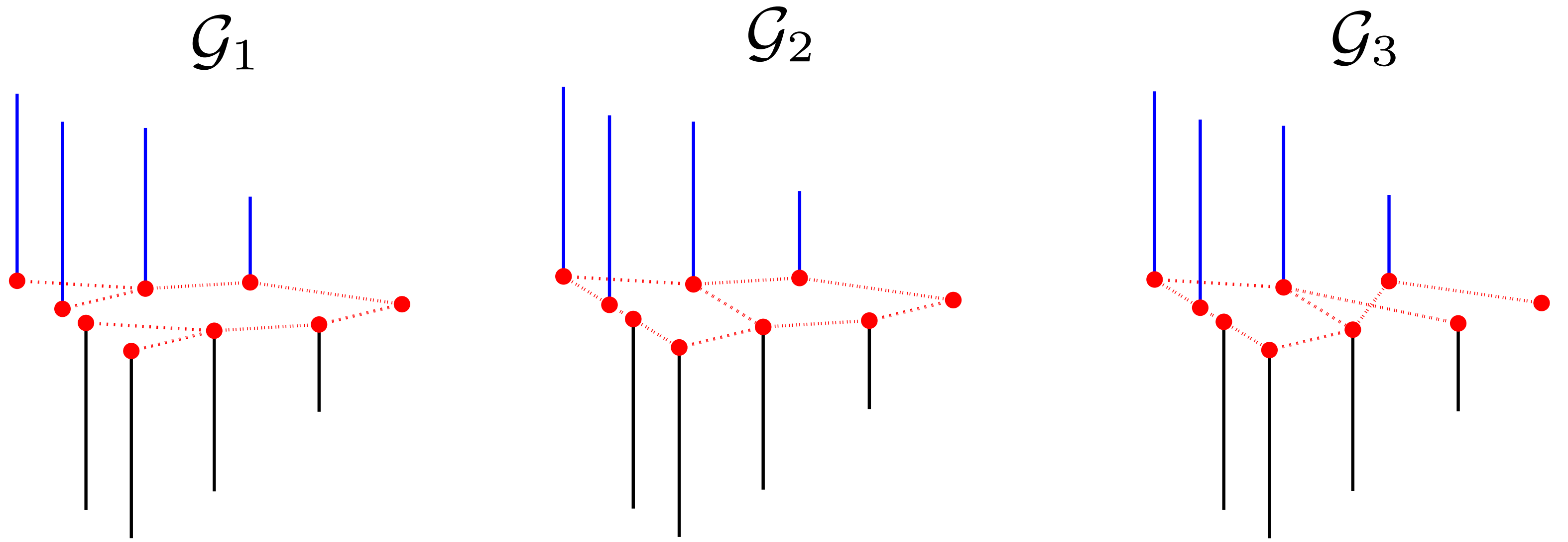
Local Variation

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

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



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

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)

Walks, Paths and Distances

Walk: a sequence of vertices $\{v_0, v_1, \ldots, v_k\}$ with $(v_{i-1}, v_i) \in E(G)$

Rem: a path is a walk with no repeating edges

Length = cardinality or sum of edge weights along path

Shortest paths and adjacency/Laplacian

$d(i,j)$ = length of shortest path between i and j

$W^n[i,j]$ = number of walks of length n between i and j

For any 2 vertices i,j if $d(i,j) > s$ then $\mathcal{L}^s[i,j] = 0$

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

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

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

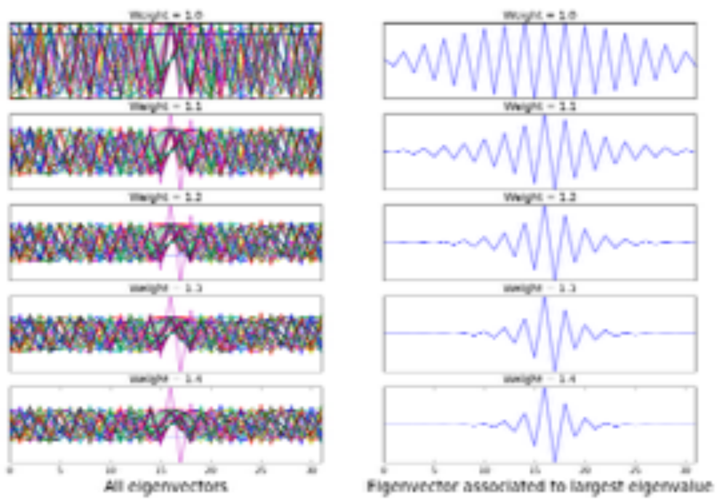
Graph Coherence

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

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

$$\min_{A\subset V} \mathrm{RatioCut}(A,\overline{A})$$

Examples: Cut and Clustering

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

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

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

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

Examples: Cut and Clustering

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

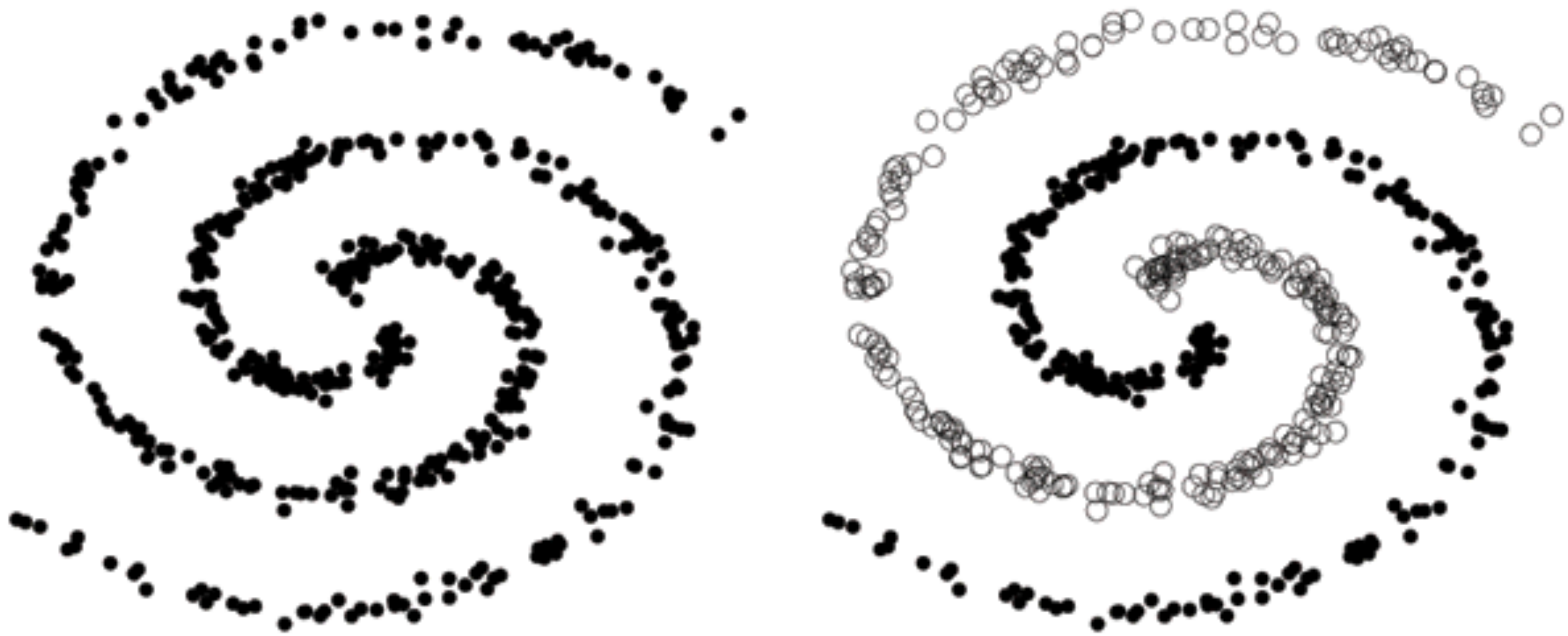
$$\min_{A\subset V} \mathrm{RatioCut}(A,\overline{A}) \qquad f[i] = \begin{cases} \sqrt{|\overline{A}|/|A|} & \text{if } i \in A \\ -\sqrt{|A|/|\overline{A}|} & \text{if } i \in \overline{A} \end{cases}$$

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

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

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

Relaxed problem Looking for a smooth partition function



Ipython Notebook example !

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

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

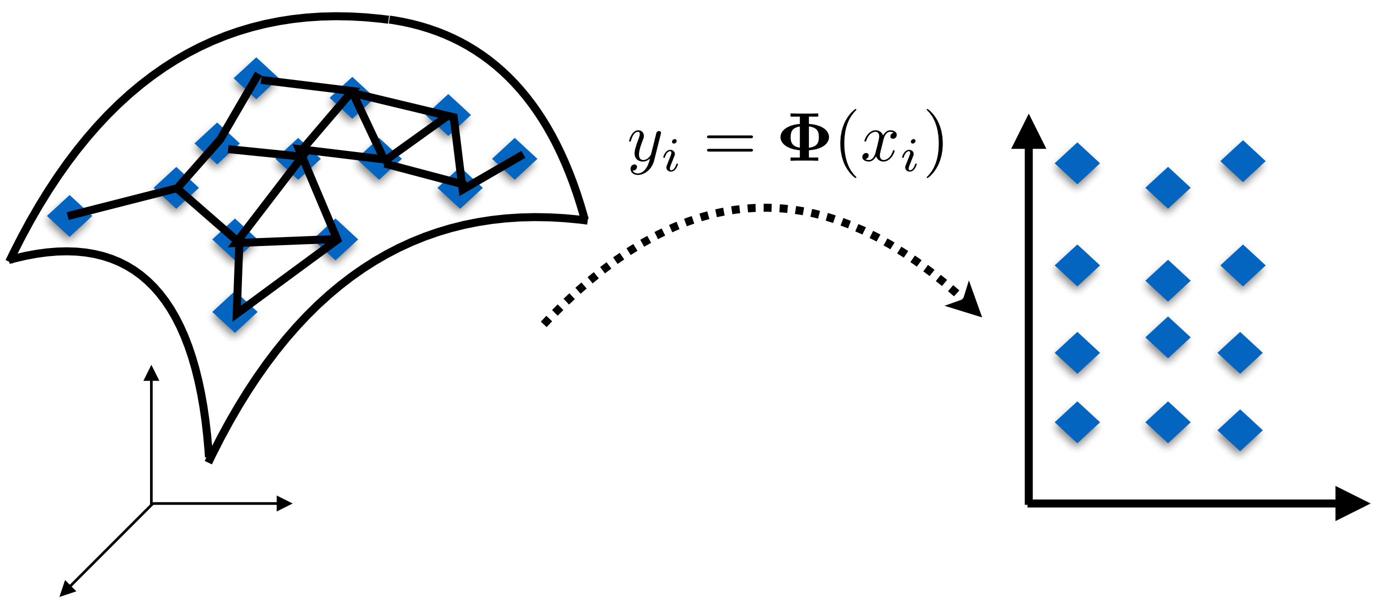
Graph Embedding/Laplacian Eigenmaps

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 $\displaystyle\sum_{i,j} W[i,j](y_i - y_j)^2$

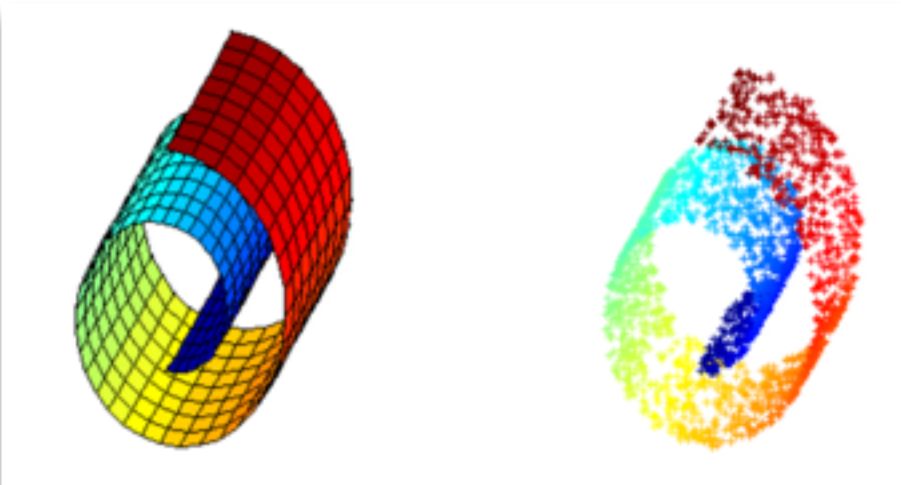
Laplacian Eigenmaps

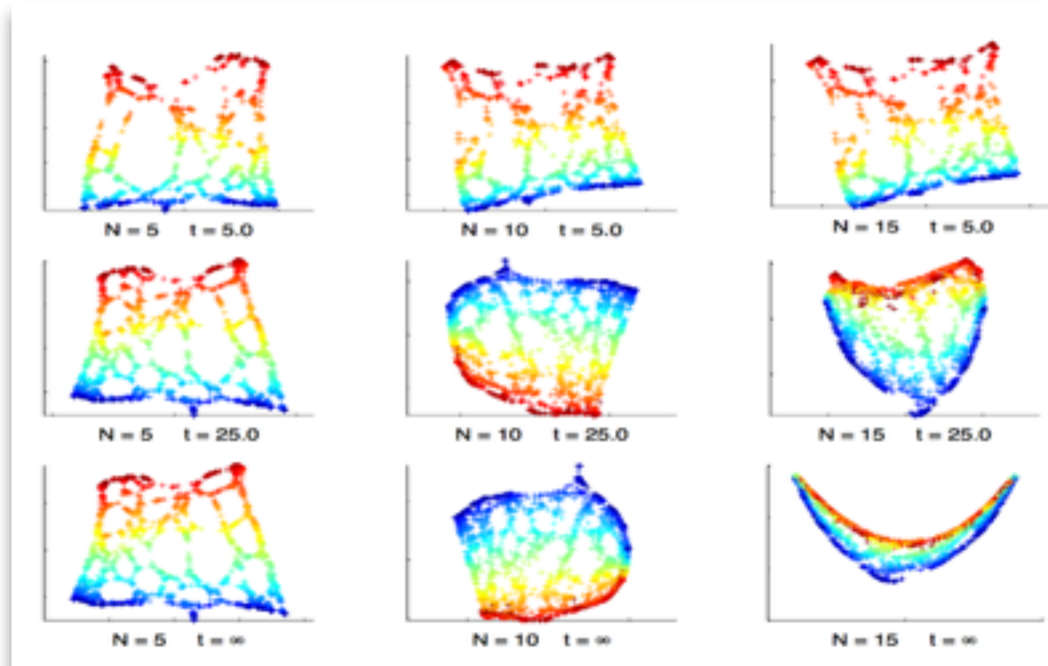
$$\arg \min_{\substack{\mathbf{y} \\ \mathbf{y}^t\mathbf{D}\mathbf{y} = 1 \\ \mathbf{y}^t\mathbf{D}\mathbf{1} = 0}} \mathbf{y}^t \mathcal{L} \mathbf{y}$$

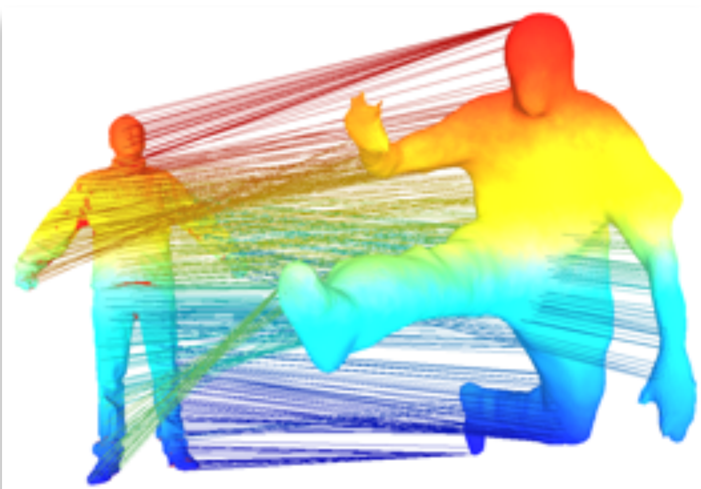
fix scale

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

Laplacian Eigenmaps







[Belkin, Niyogi, 2003]

Remark on Smoothness

Linear / Sobolev case

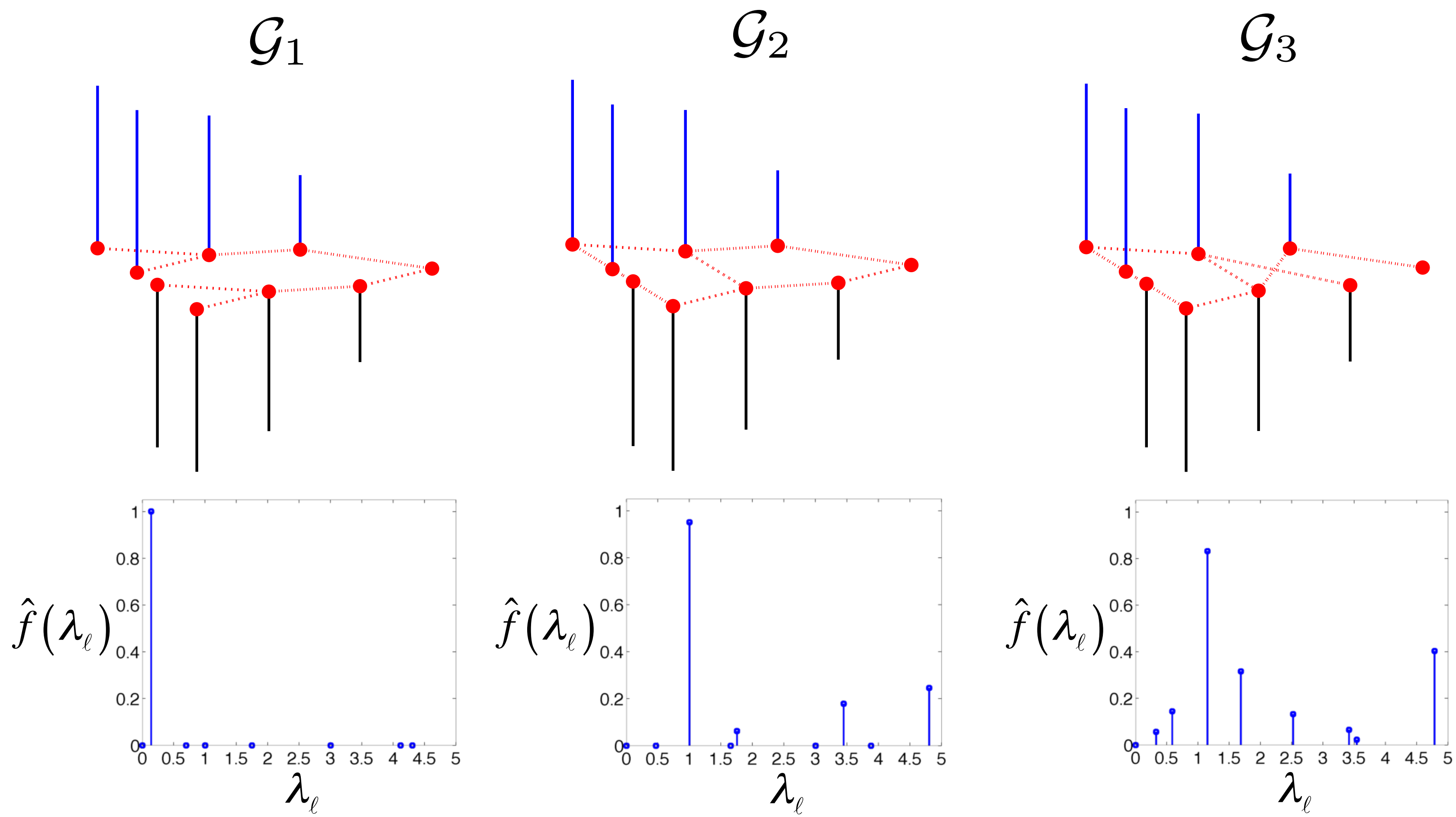
Smoothness, loosely defined, has been used to motivate various methods and algorithms. But in the discrete, finite dimensional case, asymptotic decay does not mean much

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

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

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

Smoothness of Graph Signals Revisited



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

Remark on Smoothness / Sparsity

Non-Linear / Besov Case:

$$|f|_{\mathcal{B}_p} = \Big(\sum_{k=1}^{N} |\langle \phi_k, f\rangle|^p\Big)^{1/p} \qquad 0 < p < 2$$

$$\mathcal{B}_{p,\alpha} = \Big\{ f \quad \text{s.t.}\ |f|_{\mathcal{B}_p} \leq \alpha \quad \text{with } \alpha \leq N^{1/p-1/2},\ \|f\| = 1 \Big\}$$

Best M-term approximation error: $\epsilon[M] = \sum_{k>M} |\langle \phi_{m_k}, f\rangle|^2$

Jackson-type Inequality and Sparsity

Let f $\in \mathcal{B}_{p,\alpha}$, $0 < p < 2$

$$\epsilon[M] \leq |f|^2_{\mathcal{B}_p} \tau \big(M^{-\tau} - N^{-\tau}\big) \leq \alpha^2 \tau \big(M^{-\tau} - N^{-\tau}\big)$$

with $\tau = 2/p - 1$

Functional calculus

It will be useful to manipulate functions of the Laplacian

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

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

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:

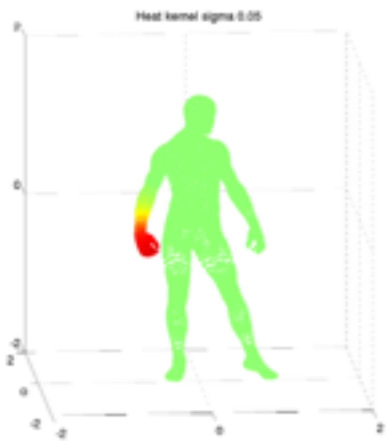
$$f(i) = \sum_{j \in V} \sum_{\ell} e^{-t\lambda_\ell} u_\ell(i) u_\ell(j) f_0(j)$$

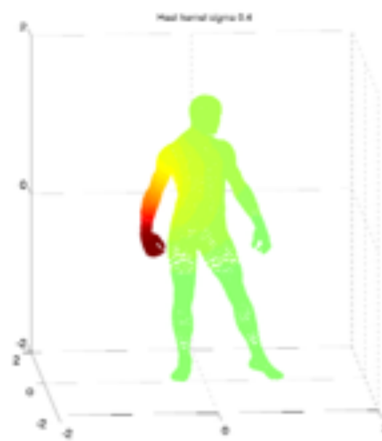
$$e^{-t\mathcal{L}} = \sum_\ell e^{-t\lambda_\ell} \mathbf{u}_\ell \mathbf{u}_\ell^t \qquad = \sum_\ell e^{-t\lambda_\ell} u_\ell(i) \sum_{j \in V} u_\ell(j) f_0(j)$$

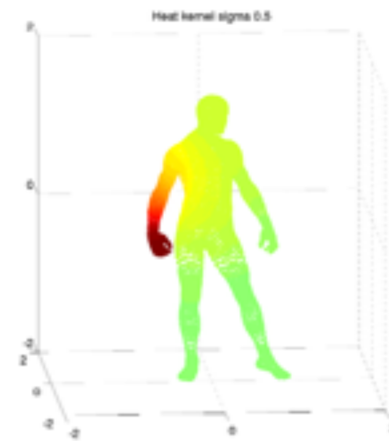
$$e^{-t\mathcal{L}}[i, j] = \sum_\ell e^{-t\lambda_\ell} u_\ell(i) u_\ell(j) \qquad = \sum_\ell e^{-t\lambda_\ell} \hat{f}_0(\ell) u_\ell(i)$$

Example: Diffusion on Graphs

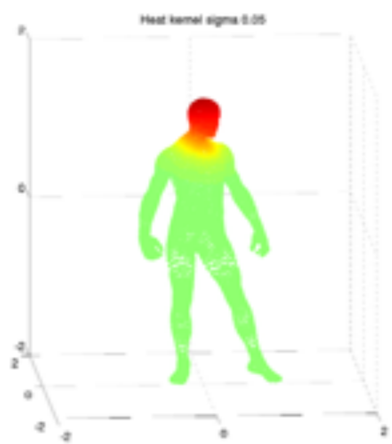
examples of heat kernel on graph







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

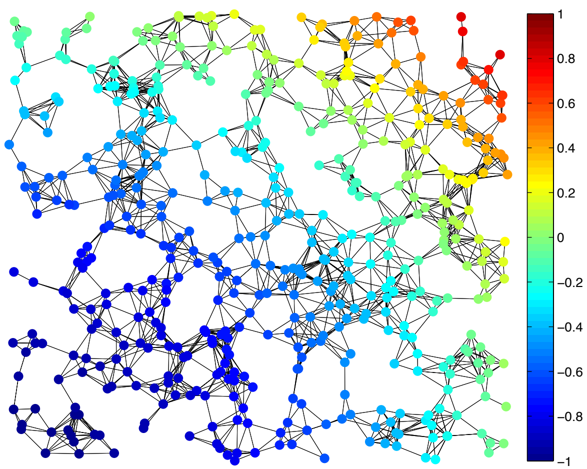


$$f(i) = \sum_{\ell} e^{-t\lambda_\ell} \hat{f}_0(\ell) u_\ell(i)$$

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

Simple De-Noising Example

Suppose a smooth signal f on a graph



Original

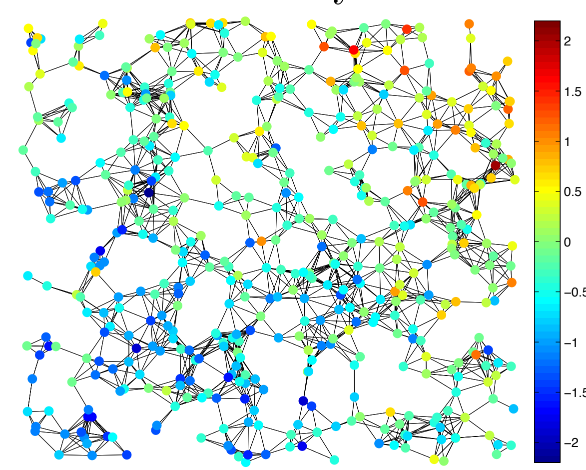
$$\|\nabla f\|_2^2 \leq M \Leftrightarrow f^t \mathcal{L} f \leq M$$

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

But you observe only a noisy version y

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

Noisy



Simple De-Noising Example

De-Noising by Regularization

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

$$\operatorname*{argmin}_f \frac{\tau}{2}\|f - y\|_2^2 + f^{\mathrm{T}} \mathcal{L}^r f$$

$$\hat{f}(\ell)\hat{g}(\lambda_\ell; \tau, r) \Rightarrow g(\mathcal{L}; \tau, r)$$

Simple De-Noising Example

De-Noising by Regularization

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

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

$$\hat{f}(\ell)\hat{g}(\lambda_\ell; \tau, r) \Rightarrow g(\mathcal{L}; \tau, r)$$

Simple De-Noising Example

De-Noising by Regularization

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

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

Graph Fourier

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

$$\hat{f}(\ell)\hat{g}(\lambda_\ell; \tau, r) \Rightarrow g(\mathcal{L}; \tau, r)$$

Simple De-Noising Example

De-Noising by Regularization

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

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

Graph Fourier

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

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

"Low pass" filtering !

$$\hat{f}(\ell)\hat{g}(\lambda_\ell; \tau, r) \Rightarrow g(\mathcal{L}; \tau, r)$$

Simple De-Noising Example

De-Noising by Regularization

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

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

Graph Fourier

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

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

"Low pass" filtering !

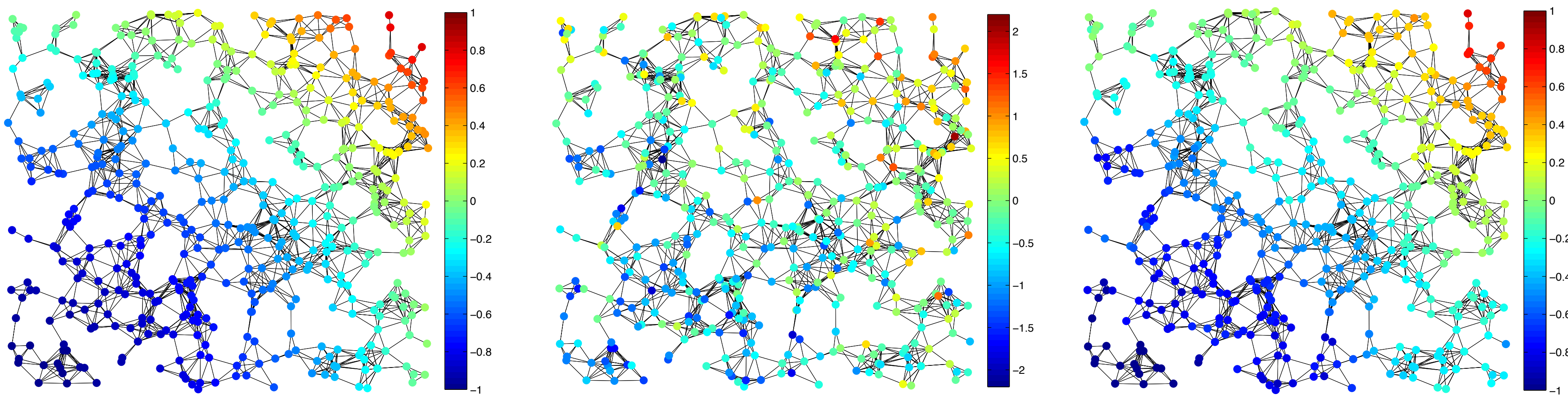
Convolution with a kernel: $\hat{f}(\ell)\hat{g}(\lambda_\ell; \tau, r) \Rightarrow g(\mathcal{L}; \tau, r)$

Simple De-Noising Example

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

Simple De-Noising Example

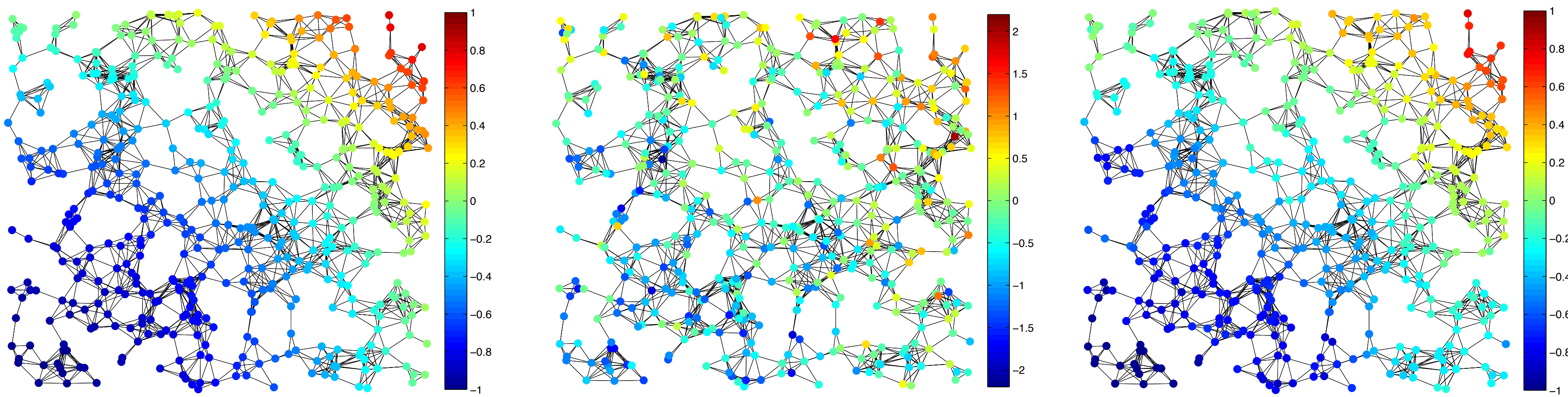
$$\operatorname{argmin}_f \left\{ ||f-y||_2^2 + \gamma f^T \mathcal{L} f \right\}$$



Original Noisy Denoised

Simple De-Noising Example

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



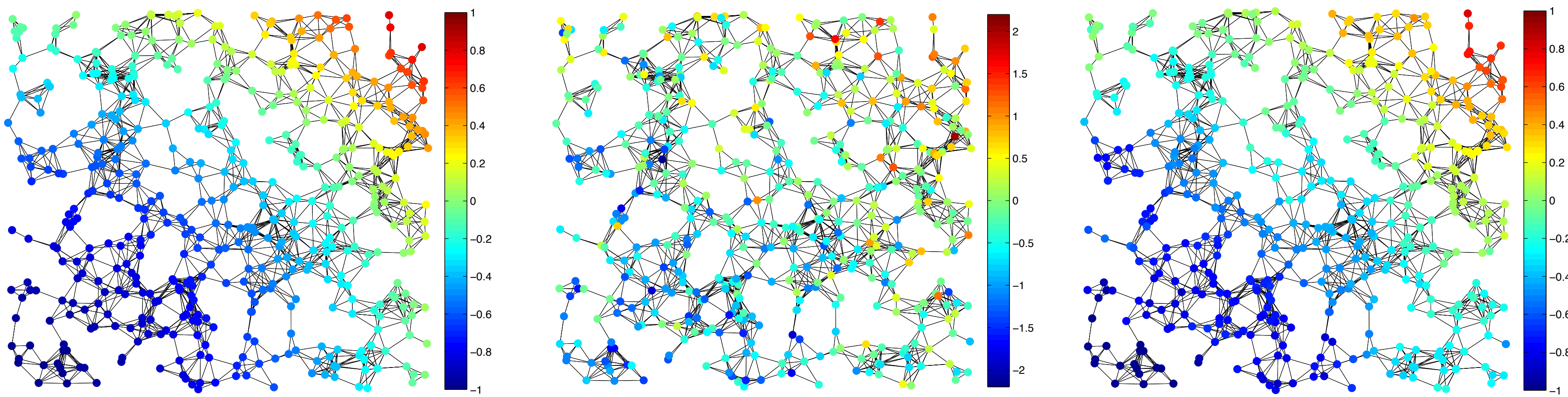
Original Noisy Denoised

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

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

Simple De-Noising Example

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



Original Noisy Denoised

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

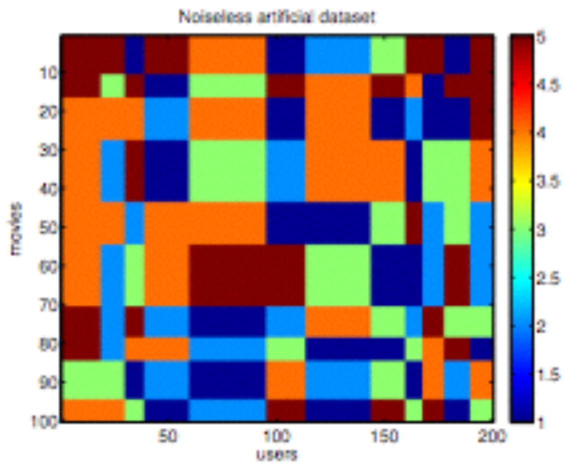
$$\Rightarrow \quad \widehat{f}_*(\ell) = \frac{\tau}{\tau + 2\lambda_\ell^r}\hat{y}(\ell) \quad \text{“Low pass” filtering !}$$

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

Application: A Recommender System

$\mathcal{L}_c$

$\mathcal{L}_r$

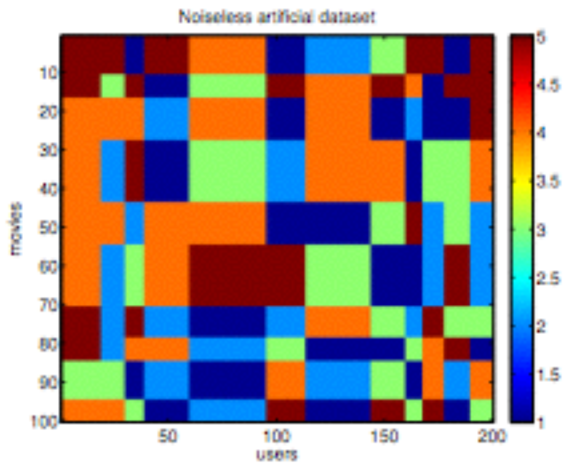


$\boldsymbol{X}[\text{movie, user}] = \text{movie rating}$

Application: A Recommender System

$\mathcal{L}_c$

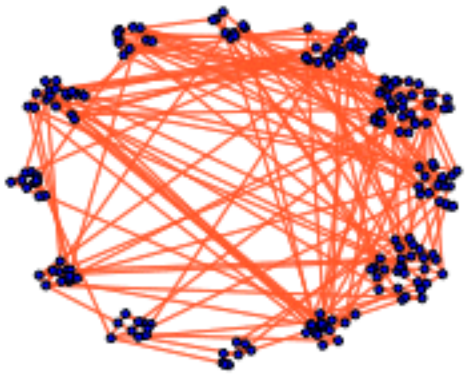
$\mathcal{L}_r$



$\boldsymbol{X}[\text{movie, user}] = \text{movie rating}$

Users structured as *communities*

(See P. Borgnat's talk)

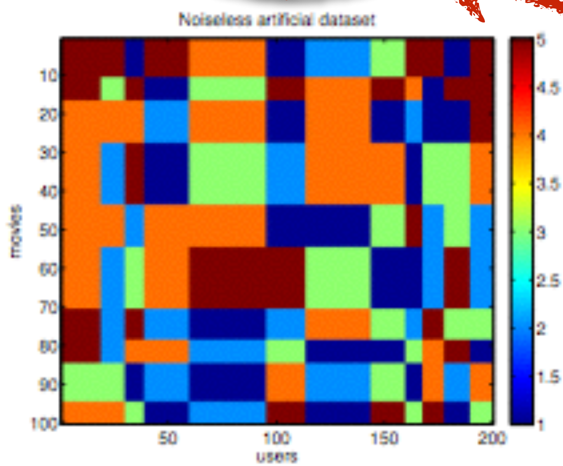


Users in community rate similarly

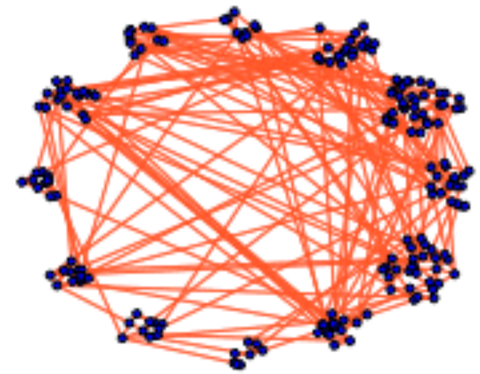
Application: A Recommender System

$\mathcal{L}_c$

$\mathcal{L}_r$



Users structured as *communities*
(See P. Borgnat's talk)

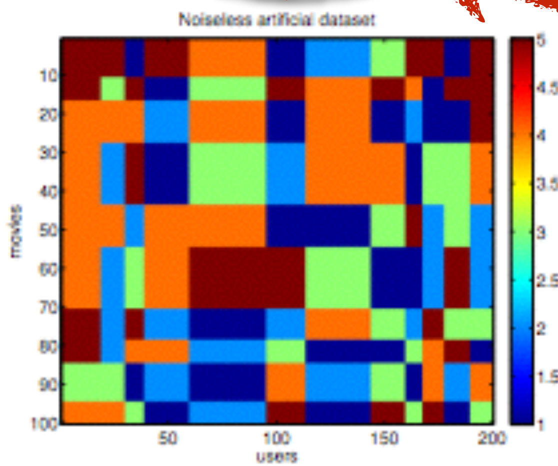


$\boldsymbol{X}[\text{movie, user}] = \text{movie rating}$

Users in community rate similarly

Application: A Recommender System

$\mathcal{L}_c$

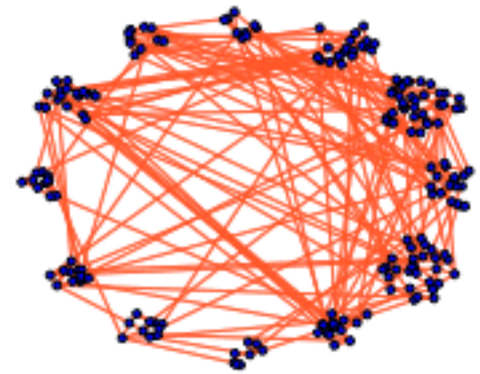


$\mathcal{L}_r$

$\boldsymbol{X}$[movie, user] $=$ movie rating

Users structured as *communities*

(See P. Borgnat's talk)



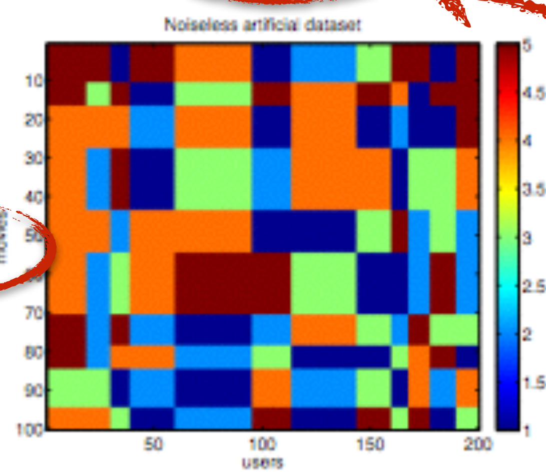
Users in community rate similarly

Movies are clustered in genres.

Similar movies are rated similarlyby users

Application: A Recommender System

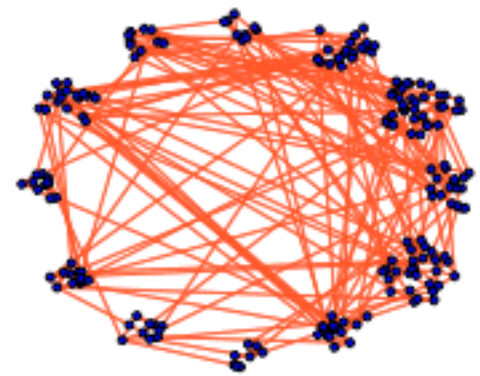
$\mathcal{L}_c$



$\mathcal{L}_r$

Users structured as *communities*

(See P. Borgnat's talk)



$\boldsymbol{X}$[movie, user] $=$ movie rating

Users in community rate similarly

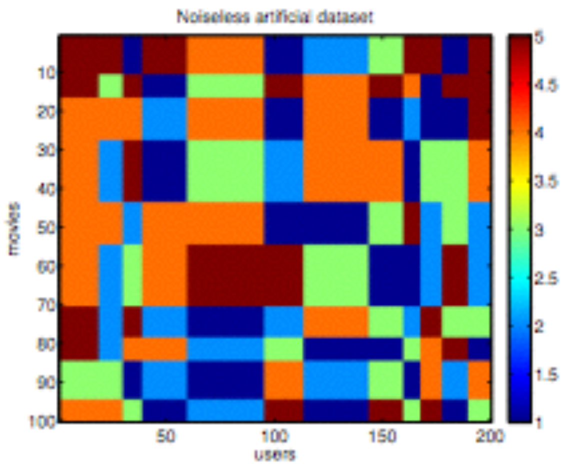
Movies are clustered in genres.

Similar movies are rated similarlyby users

Application: A Recommender System

$\mathcal{L}_c$

$\mathcal{L}_r$

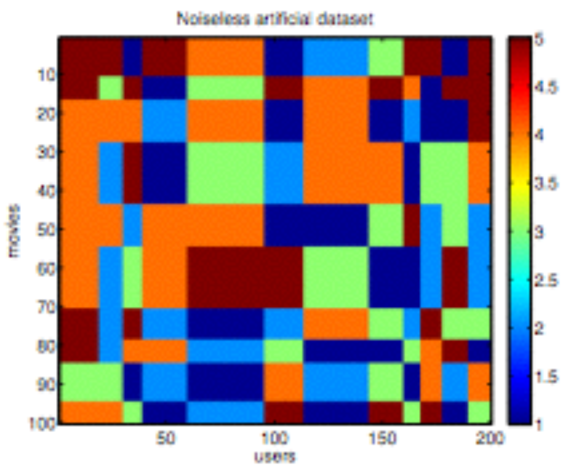


$\boldsymbol{X}[\text{movie, user}] = \text{movie rating}$

Application: A Recommender System

$\mathcal{L}_c$

$\mathcal{L}_r$



$\boldsymbol{X}[\text{movie, user}] = \text{movie rating}$

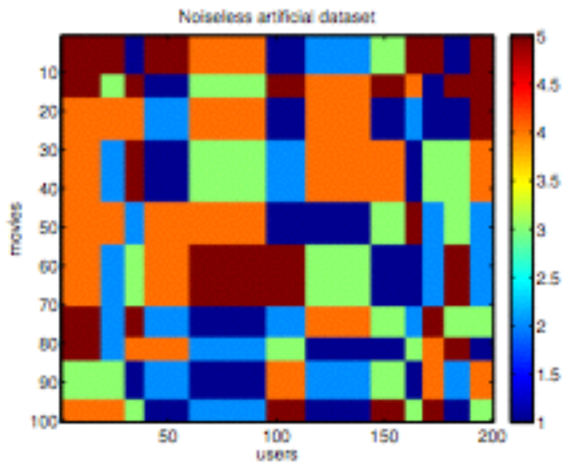
$$\arg\min_{\mathbf{X}} \gamma_n \|\mathbf{X}\|_* + \|A_\Omega \circ (\mathbf{X} - \mathbf{M})\| + \gamma_r \mathbf{X}\mathcal{L}_r\mathbf{X}^t + \gamma_c \mathbf{X}^t\mathcal{L}_c\mathbf{X}$$

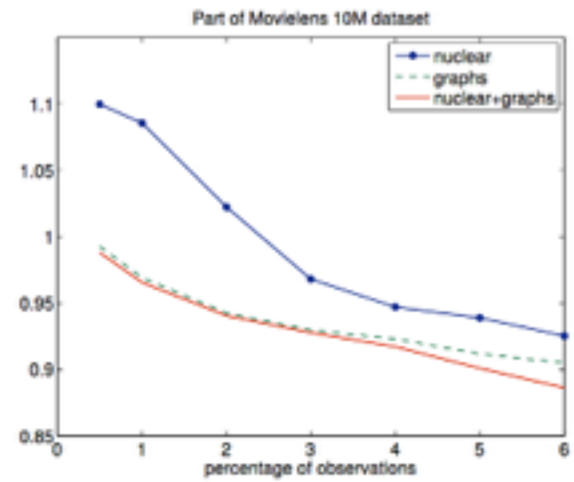
Solved using ADMM

Application: A Recommender System

$\mathcal{L}_c$

$\mathcal{L}_r$





$\boldsymbol{X}[\text{movie, user}] = \text{movie rating}$

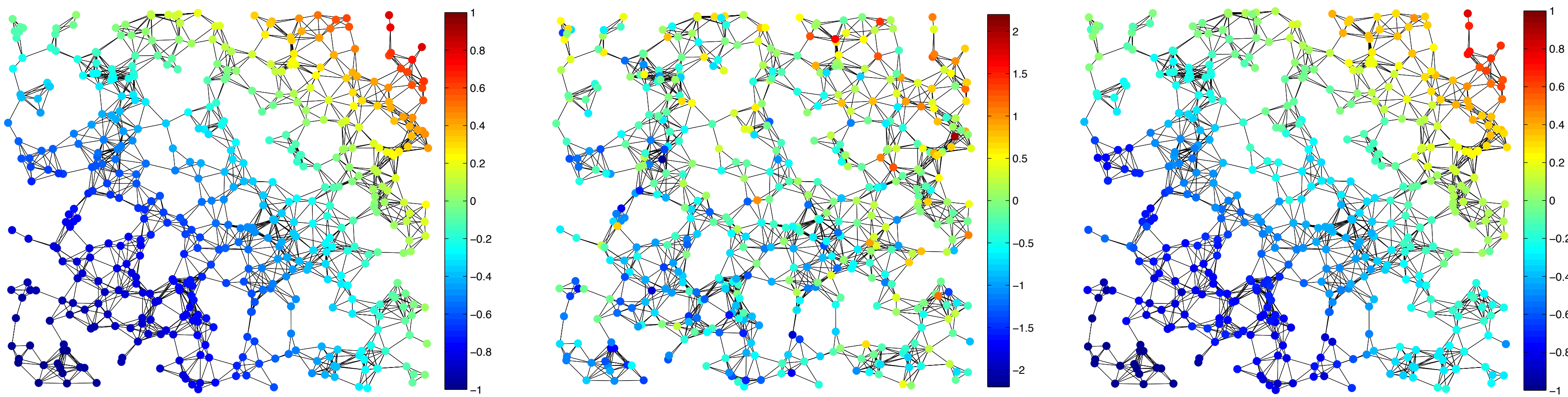
$$\arg\min_{\mathbf{X}} \gamma_n \|\mathbf{X}\|_* + \|A_\Omega \circ (\mathbf{X} - \mathbf{M})\| + \gamma_r \mathbf{X}\mathcal{L}_r\mathbf{X}^t + \gamma_c \mathbf{X}^t\mathcal{L}_c\mathbf{X}$$

Solved using ADMM

Convolution with a kernel and localization

Simple De-Noising Example

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



Original Noisy Denoised

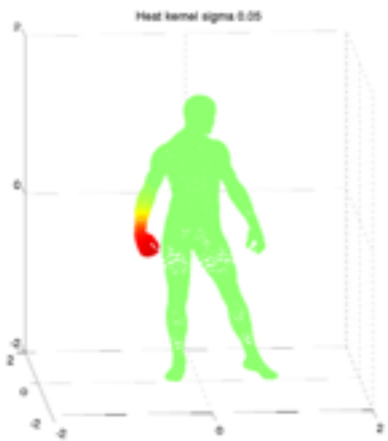
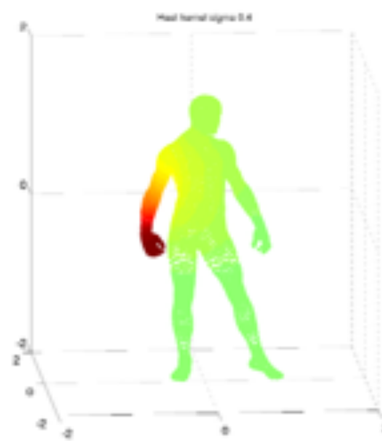
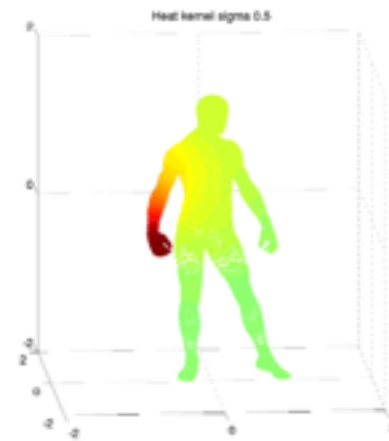
$$\underset{f}{\mathrm{argmin}}\ \frac{\tau}{2}\|f - y\|_2^2 + f^{\mathrm{T}} \mathcal{L}^r f \quad \Longrightarrow \quad \mathcal{L}^r f_* + \frac{\tau}{2}(f_* - y) = 0$$

$$\Longrightarrow \quad \widehat{f}_*(\ell) = \frac{\tau}{\tau + 2\lambda_\ell^r}\hat{y}(\ell) \quad \text{“Low pass” filtering !}$$

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

Example: Diffusion on Graphs

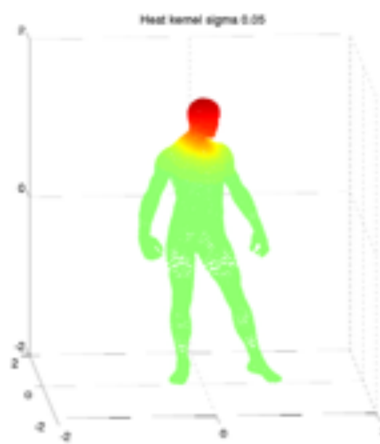
examples of heat kernel on graph

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

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

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



"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

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

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

Use convolution to induce translations

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

Spectral Graph Wavelets

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

Spectral Graph Wavelets

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

- Generalized translation

Spectral Graph Wavelets

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

- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$

Spectral Graph Wavelets

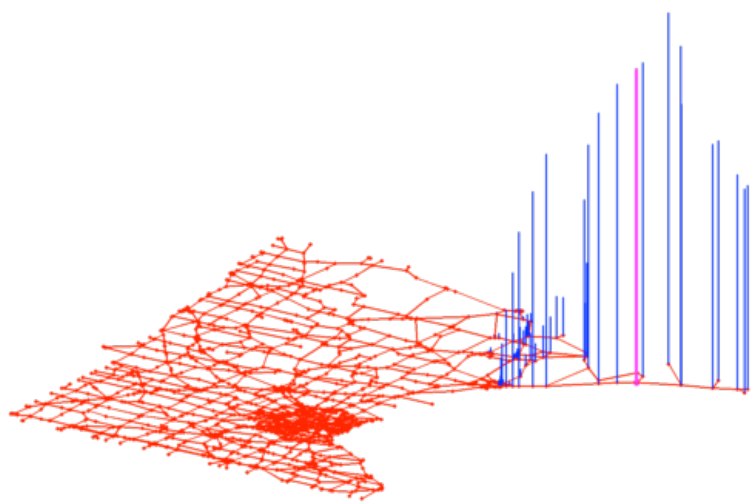
Hammond et al., Wavelets on graphs via spectral graph theory, ACHA, 2011

- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$

Spectral Graph Wavelets

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

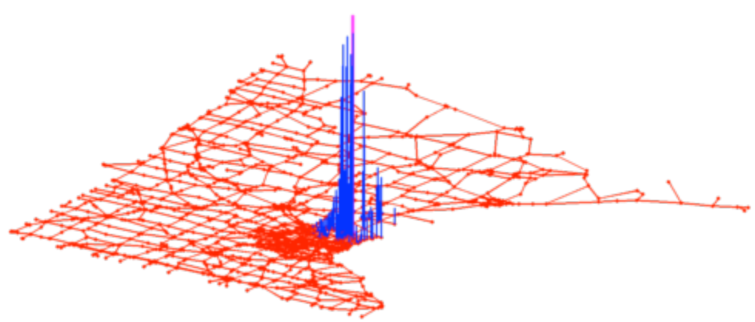
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



Spectral Graph Wavelets

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

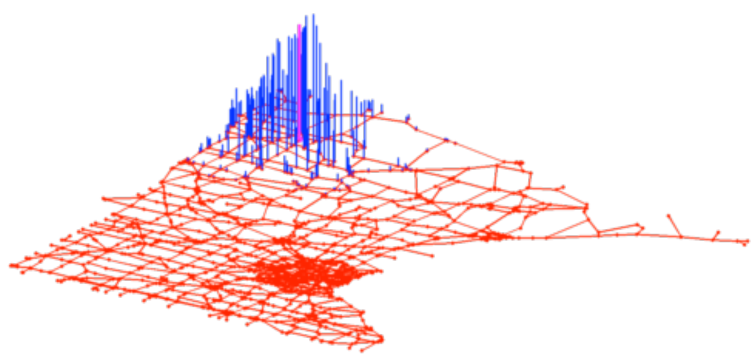
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



Spectral Graph Wavelets

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

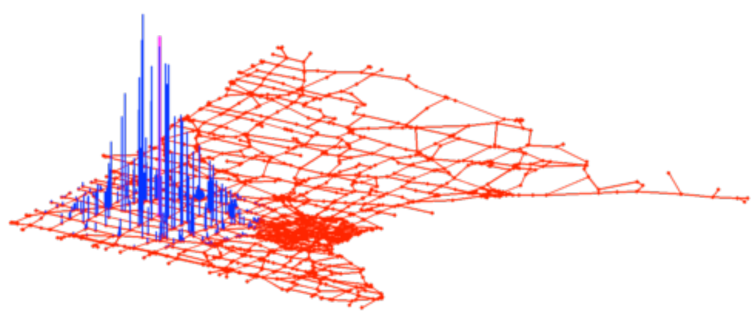
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



Spectral Graph Wavelets

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

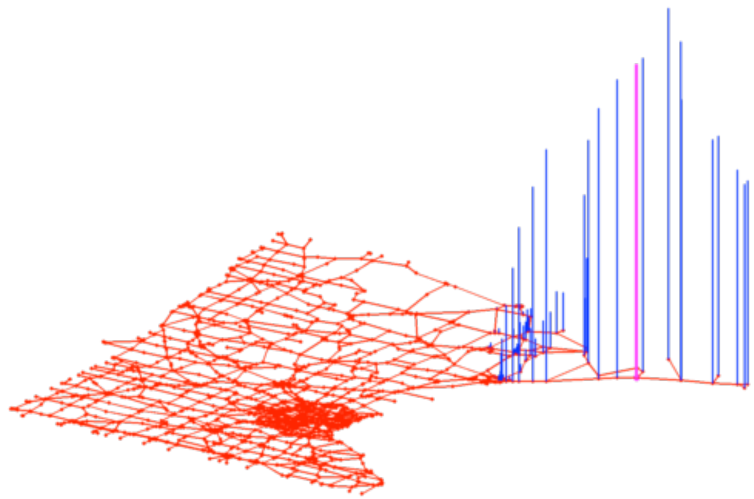
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



Spectral Graph Wavelets

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

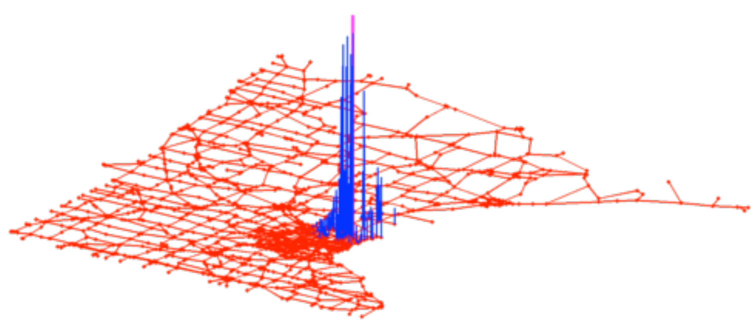
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



Spectral Graph Wavelets

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

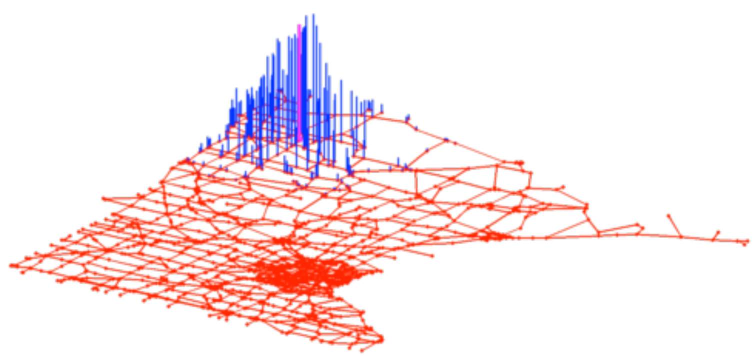
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



Spectral Graph Wavelets

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

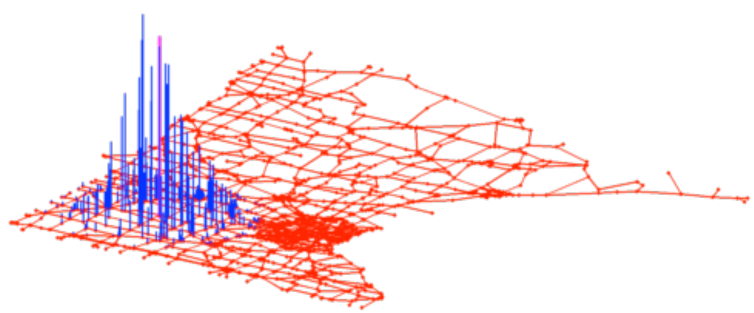
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



Spectral Graph Wavelets

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

- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



The Agonizing Limits of Intuition

The Graph Fourier and Kronecker bases are not necessarily mutually unbiased

$$\mu := \max_{\substack{\ell \in \{0,1,\dots,N-1\} \\ i \in \{1,2,\dots,N\}}} |\langle \chi_\ell, \delta_i \rangle| \in \left[\frac{1}{\sqrt{N}}, 1 \right]$$

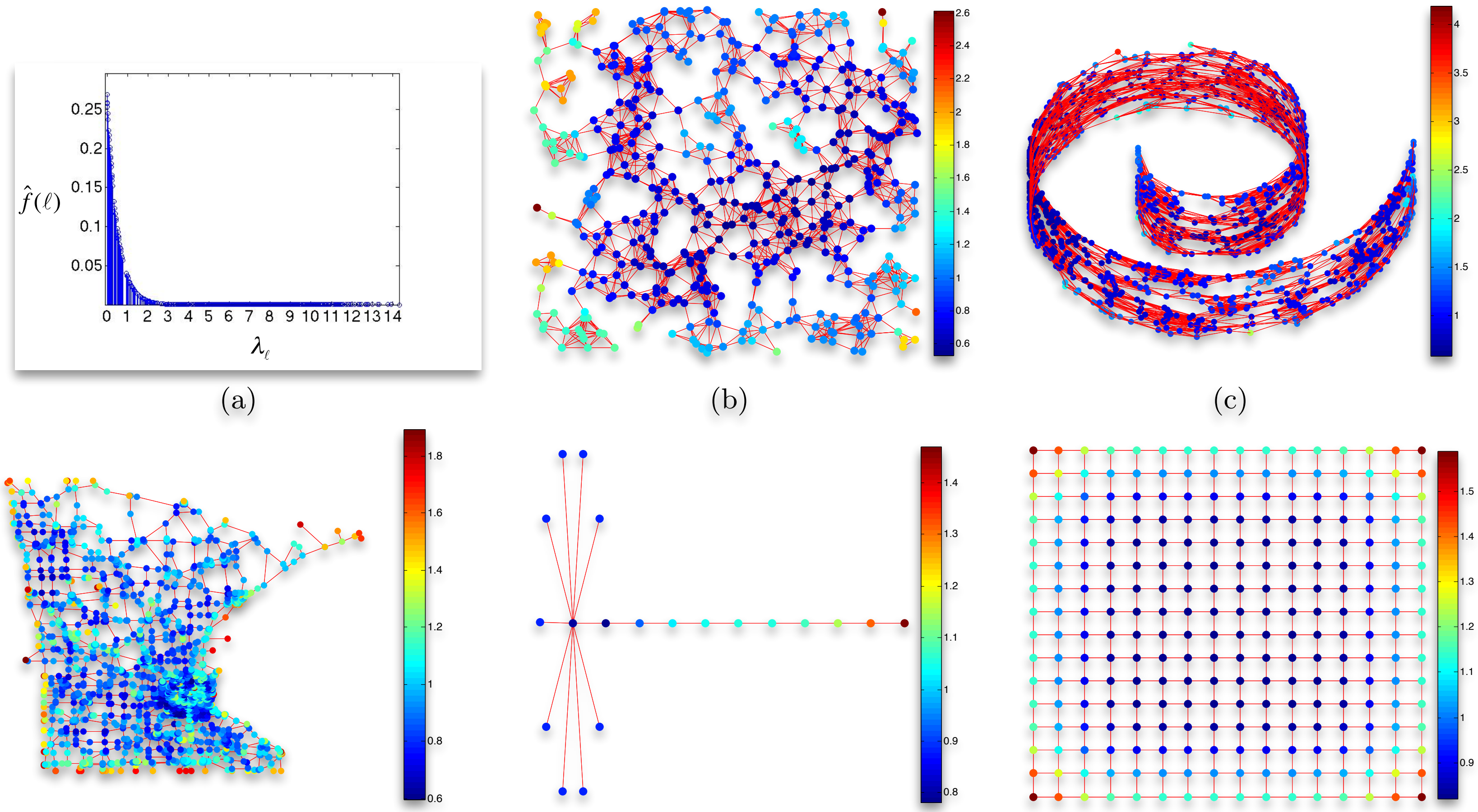
Laplacian eigenvectors (Fourier modes!) can be well localized

- 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:

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

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:

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

Are these features localized ?

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

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

Should be well localized within *K*-ball around n !

Polynomial Localization

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

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

Are these features localized ?

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

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

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

Should be well localized within *K*-ball around n !

Polynomial Localization

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

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

Are these features localized ?

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

Construct an order K polynomial approximation:

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

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

Should be well localized within K-ball around n !

Polynomial Localization

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

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

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

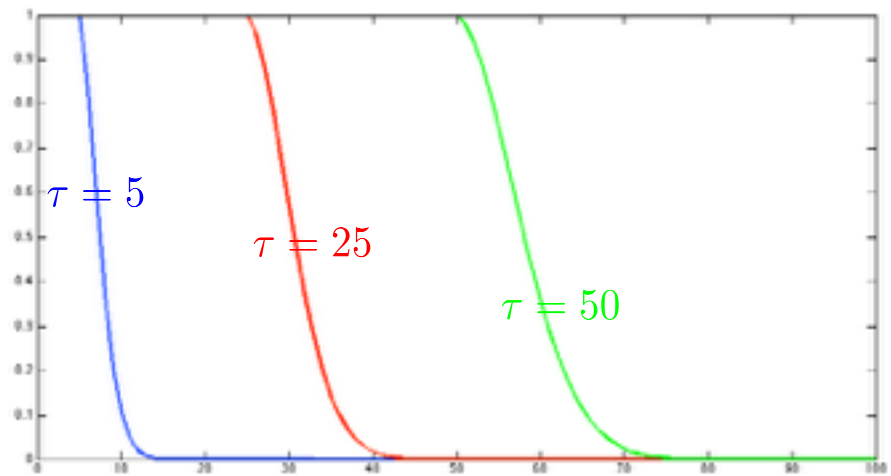
Example: for the heat kernel $\hat{g}(\lambda) = e^{-\tau\lambda}$

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

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

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

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



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

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

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Localization / Uncertainty

Localization / Uncertainty

Competition between smoothness and localization in the spectral representation of kernels

Localization / Uncertainty

Competition between smoothness and localization in the spectral representation of kernels

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

Localization / Uncertainty

Competition between smoothness and localization in the spectral representation of kernels

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

Smooth kernels can be used to construct controlled localized features

Example: Spectral Graph Wavelets

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)

Spectral approaches to multiresolution

Spectral Graph Wavelets

Remember good old Euclidean case:

$$(W^s f)(x) = \frac{1}{2\pi} \int_{\mathbf{R}} e^{i\omega x} \hat{\psi}^*(s\omega) \hat{f}(\omega) d\omega$$

We will adopt this operator view

$$\hat{g} : \mathbb{R}^+ \mapsto \mathbb{R} \qquad W_g = g(\mathcal{L})$$

$$\widehat{W_g f}(\ell) = \hat{g}(\lambda_\ell) \hat{f}(\ell) \qquad (W_g f)(i) = \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) \hat{f}(\ell) u_\ell(i)$$

Spectral Graph Wavelets

Remember good old Euclidean case:

$$(W^s f)(x) = \frac{1}{2\pi} \int_{\mathbf{R}} e^{i\omega x} \hat{\psi}^*(s\omega) \hat{f}(\omega) d\omega$$

We will adopt this operator view

Operator-valued function via continuous *Borel functional calculus*

$$\hat{g} : \mathbb{R}^+ \mapsto \mathbb{R} \qquad W_g = g(\mathcal{L})$$ **Operator-valued function**

Action of operator is induced by its Fourier symbol

$$\widehat{W_g f}(\ell) = \hat{g}(\lambda_\ell) \hat{f}(\ell) \qquad \big(W_g f\big)(i) = \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) \hat{f}(\ell) u_\ell(i)$$

Spectral Graph Wavelets

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

Generalized translation

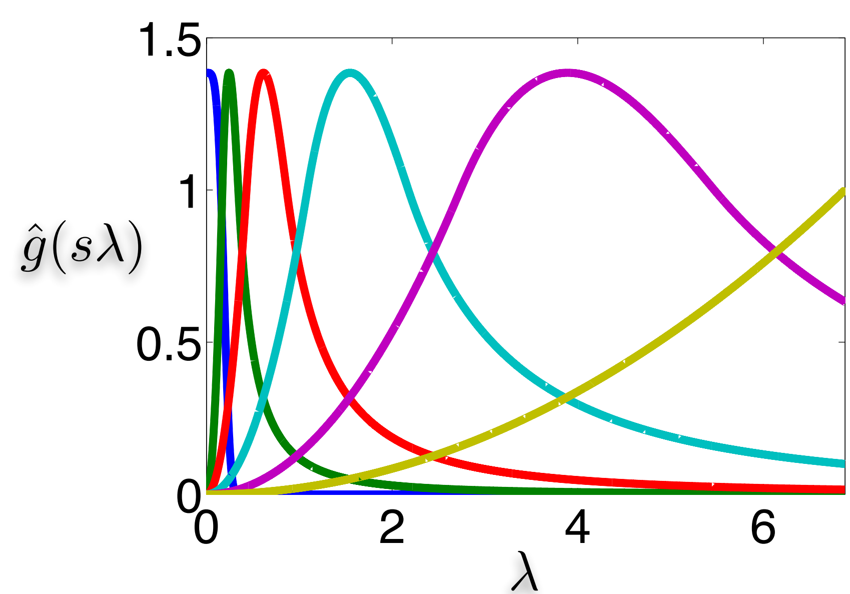
- Classical setting:

$$(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$$

- Graph setting:

$$(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$$

- Generalized dilation: $\widehat{\mathcal{D}_s g}(\lambda) = \hat{g}(s\lambda)$



Spectral Graph Wavelets

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

Generalized translation

- Classical setting:

$$(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$$

- Graph setting:

$$(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$$

- Generalized dilation: $\widehat{\mathcal{D}_s g}(\lambda) = \hat{g}(s\lambda)$

- Spectral graph wavelet at scale s, centered at vertex n:

$$\boxed{\psi_{s,n}(i) := (T_n D_s g)(i) = \sum_{\ell=0}^{N=1} \hat{g}(s\lambda_\ell) u_\ell^*(n) u_\ell(i)}$$

Spectral Graph Wavelets

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

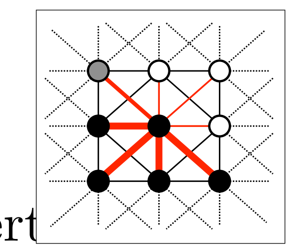
- Generalized translation
  - Classical setting:

$$(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$$

  - Graph setting:

$$(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$$

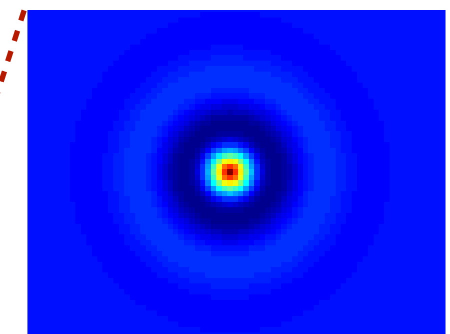
Semi-Local Graph

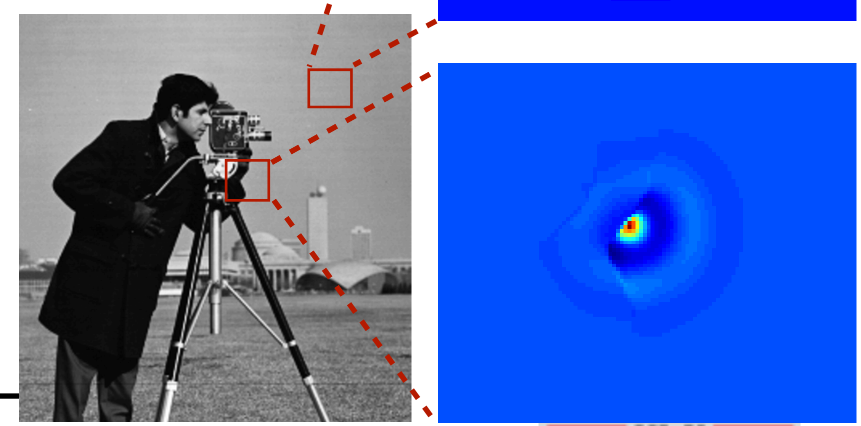


- Generalized dilation:

$$\widehat{\mathcal{D}_s g}(\lambda) = \hat{g}(s\lambda)$$

- Spectral graph wavelet at scale s, centered at vert





$$\boxed{\psi_{s,n}(i) := (T_n D_s g)(i) = \sum_{\ell=0}^{N=1} \hat{g}(s\lambda_\ell) u_\ell^*(n) u_\ell(i)}$$

Spectral Graph Wavelets

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

Generalized translation

- Classical setting:

$$(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$$

- Graph setting:

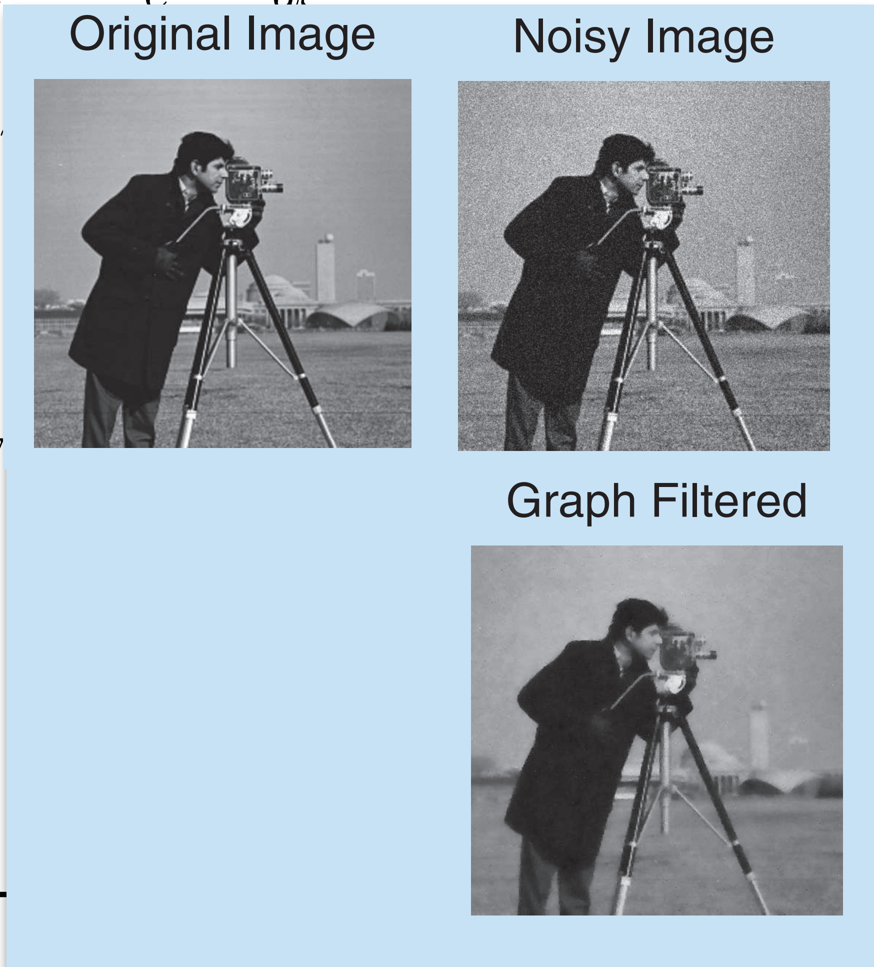
$$(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell($$

- Generalized dilation:

$$\widehat{D_s g}(\lambda) = \hat{g}(s\lambda)$$

- Spectral graph wavelet at scale s, centered at v

$$\boxed{\psi_{s,n}(i) := (T_n D_s g)(i) = \sum_{\ell=0}^{N=1} \hat{g}(s\lambda_\ell) u_\ell^*(n) u_\ell(i)}$$



ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

A Continuous Wavelet Transform

Continuous Spectral Graph Wavelet Transform

$$\big(W_g f\big)(t,j) = \big(g(t\mathcal{L})f\big)(j) = \sum_{\ell} \hat{g}(t\lambda_\ell)\hat{f}(\ell)u_\ell(j)$$

If kernel satisfies $$C_g = \int_0^{+\infty} \frac{\hat{g}^2(x)}{x} < +\infty$$

Inverse Transform

$$\frac{1}{C_g}\sum_{j\in V}\int_0^{+\infty} W_g f(t,j)\psi_{t,j}(i)\frac{dt}{t} = \tilde{f}(i) \qquad \tilde{f} = f - \langle u_0, f\rangle u_0$$

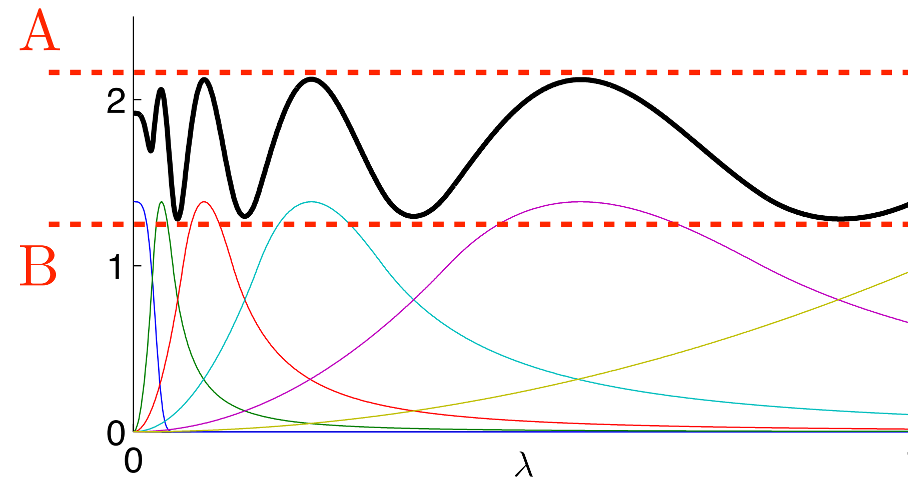
Frames

$\exists A, B > 0,\ \exists \hat{h} : \mathbb{R}^+ \mapsto \mathbb{R}$ i.e scaling function

$$0 < A \leq \hat{h}(u)^2 + \sum_s \hat{g}(t_s u)^2 \leq B < +\infty$$

scaling function *wavelets*

$\phi_n = W_h \delta_n = h(\mathcal{L})\delta_n$



A simple way to get a tight frame:

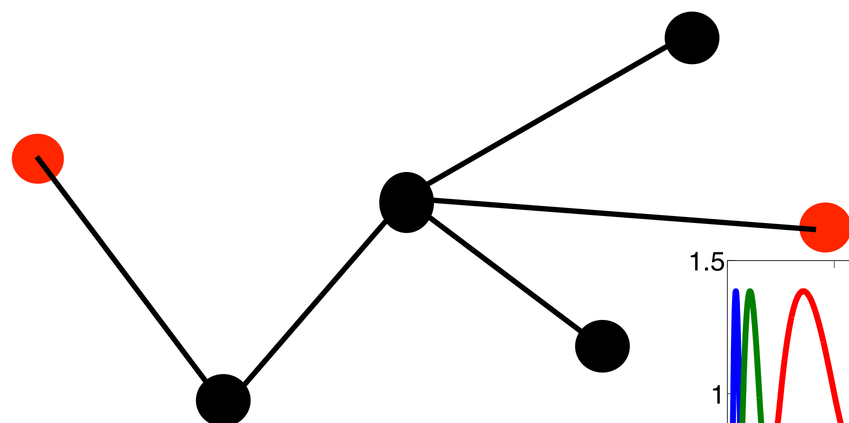
$$\hat{\gamma}(\lambda_\ell) = \int_{1/2}^{1} \frac{dt}{t} \hat{g}(t\lambda_\ell)^2 \Rightarrow \tilde{\hat{g}}(\lambda_\ell) = \sqrt{\hat{\gamma}(\lambda_\ell) - \hat{\gamma}(2\lambda_\ell)}$$

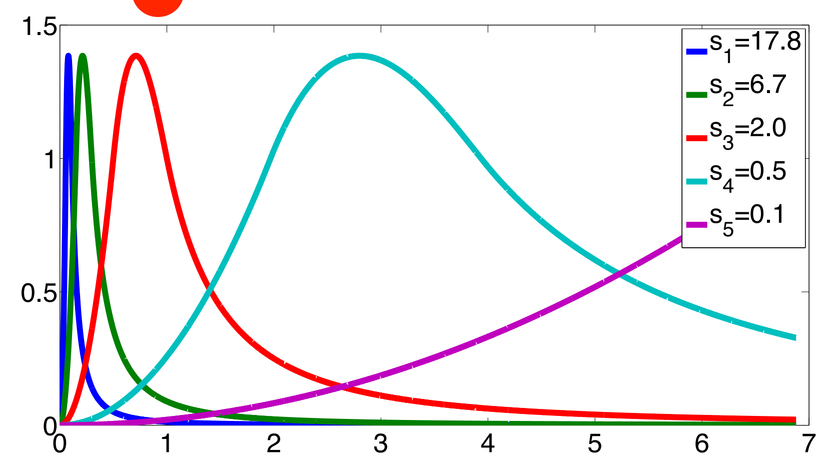
for any admissible kernel

Scaling & Localization

Effect of operator dilation ?

Need higher polynomial approximation for large scale kernel (on spectral domain)!

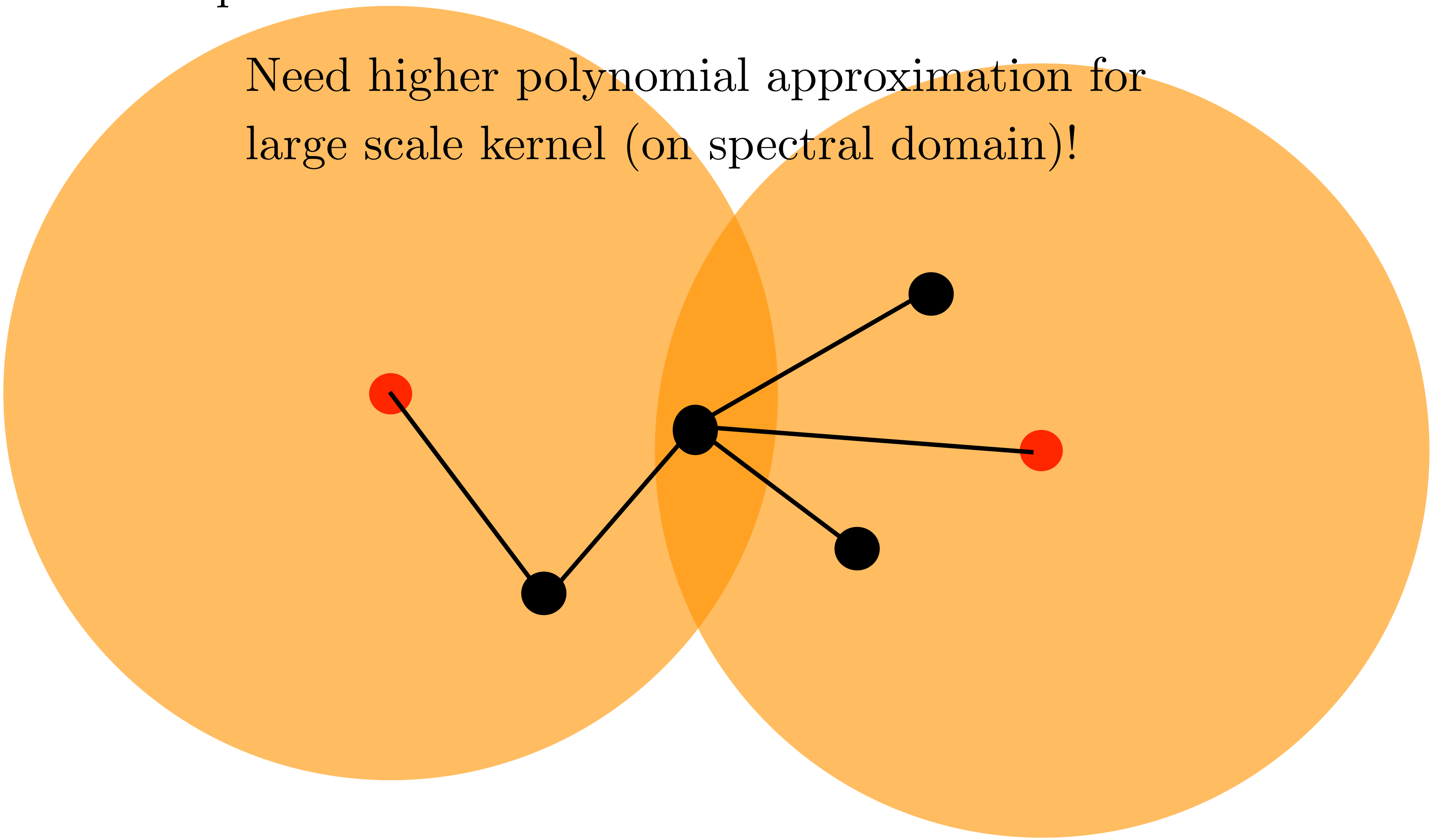




Scaling & Localization

Effect of operator dilation ?

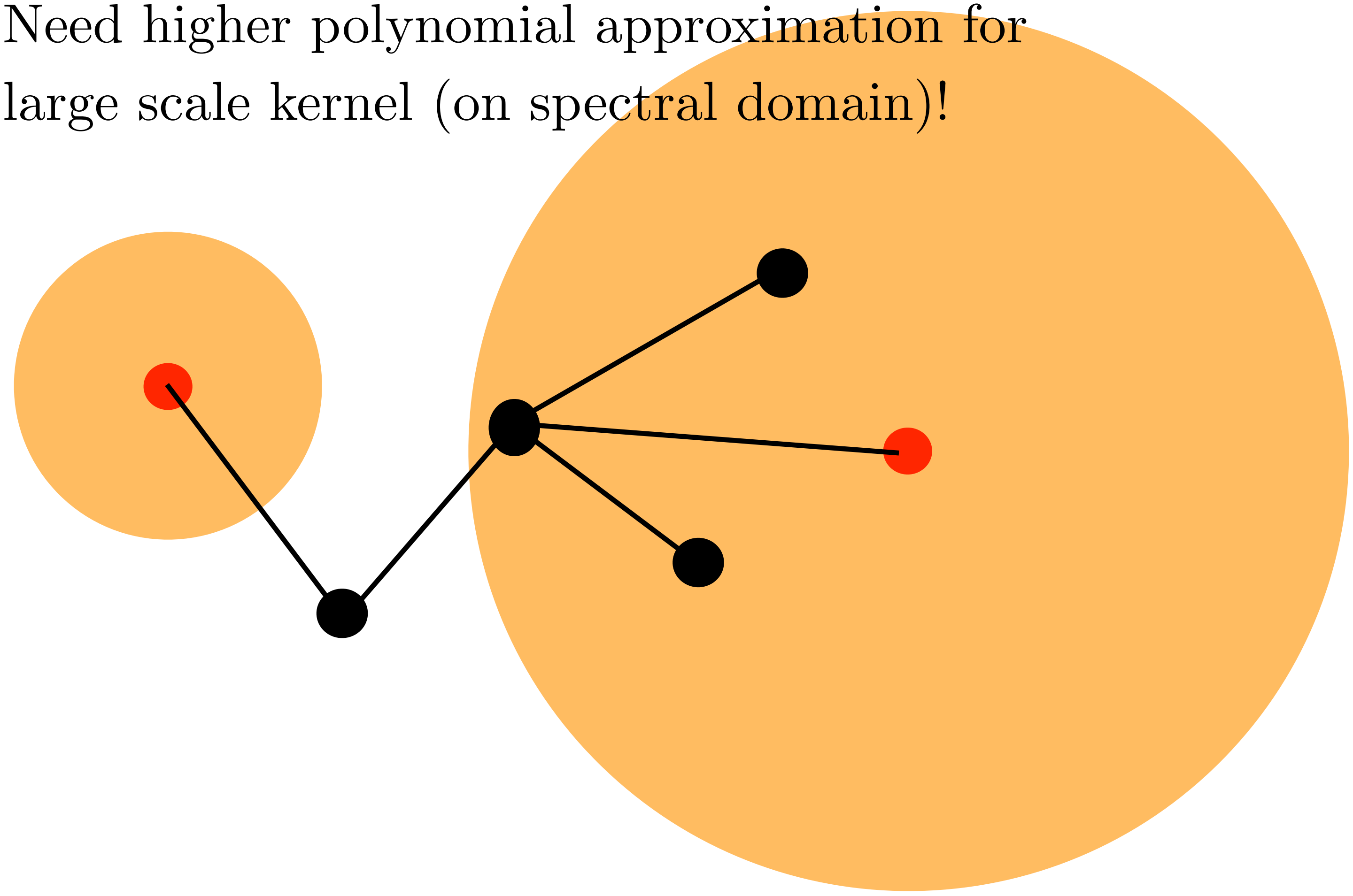
Need higher polynomial approximation for large scale kernel (on spectral domain)!



Scaling & Localization

Effect of operator dilation ?

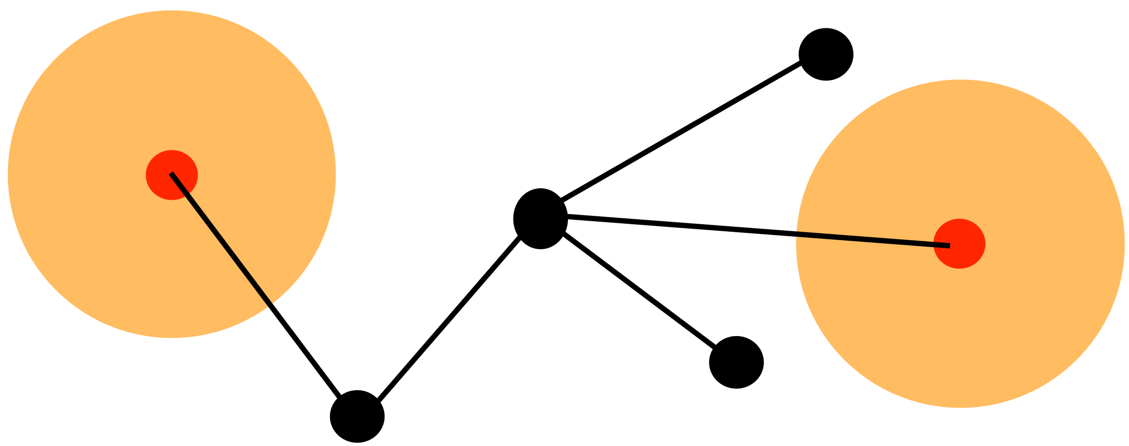
Need higher polynomial approximation for large scale kernel (on spectral domain)!



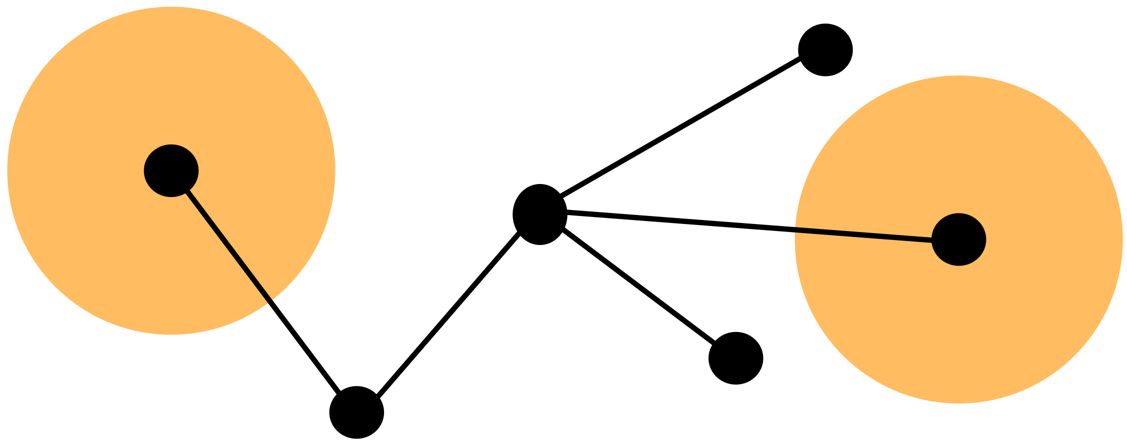
Scaling & Localization

Effect of operator dilation ?

Need higher polynomial approximation for large scale kernel (on spectral domain)!



Scaling & Localization



$\psi_{t,i}(j)$ should be small if i and j are separated, and t is small

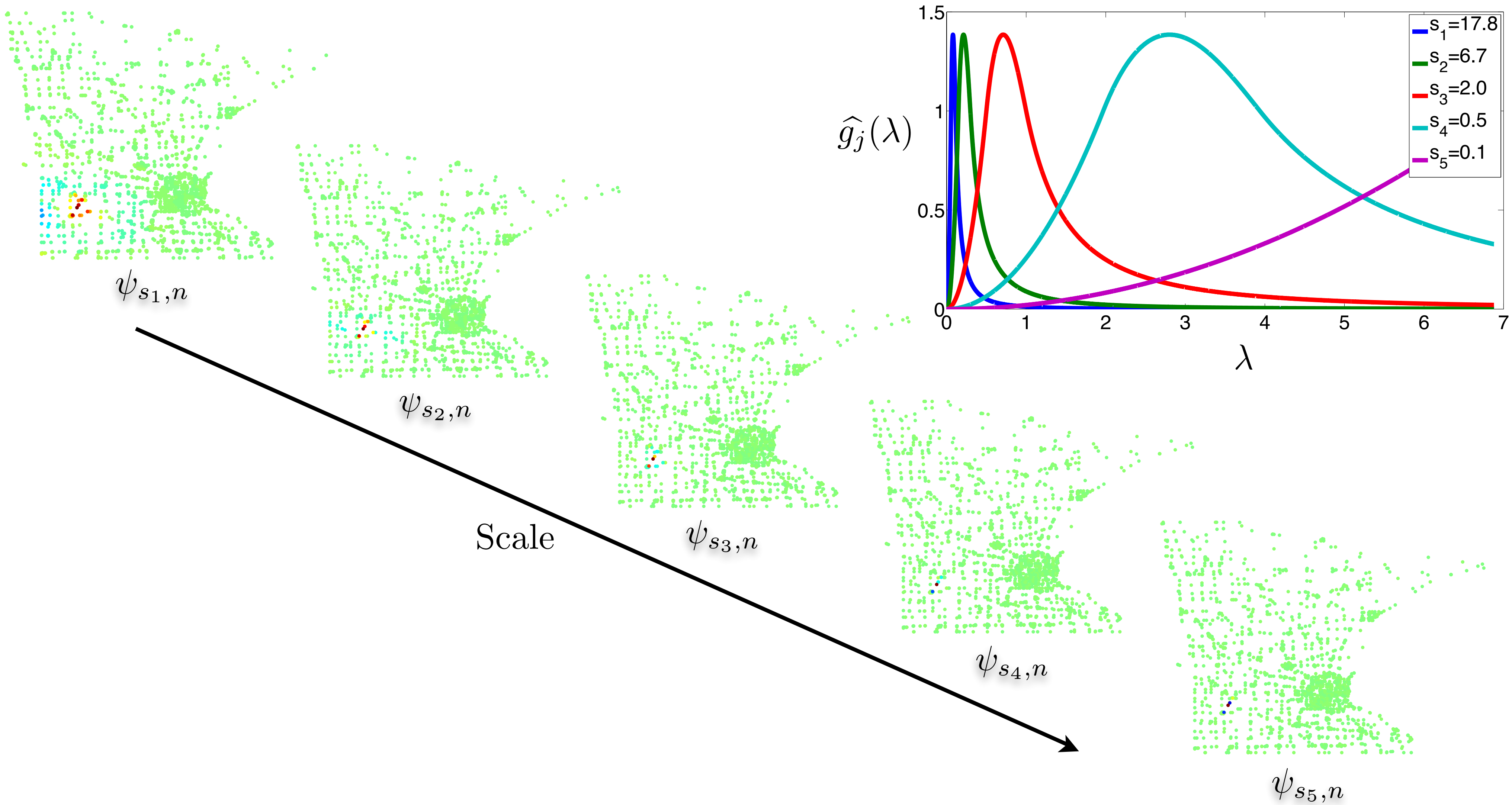
Study matrix element: $\psi_{t,i}(j) = \langle \psi_{t,i}, \delta_j \rangle = \langle T_g^t \delta_i, \delta_j \rangle$

Theorem: $d_G(i,j) > K$ and g has K vanishing derivatives at *0*

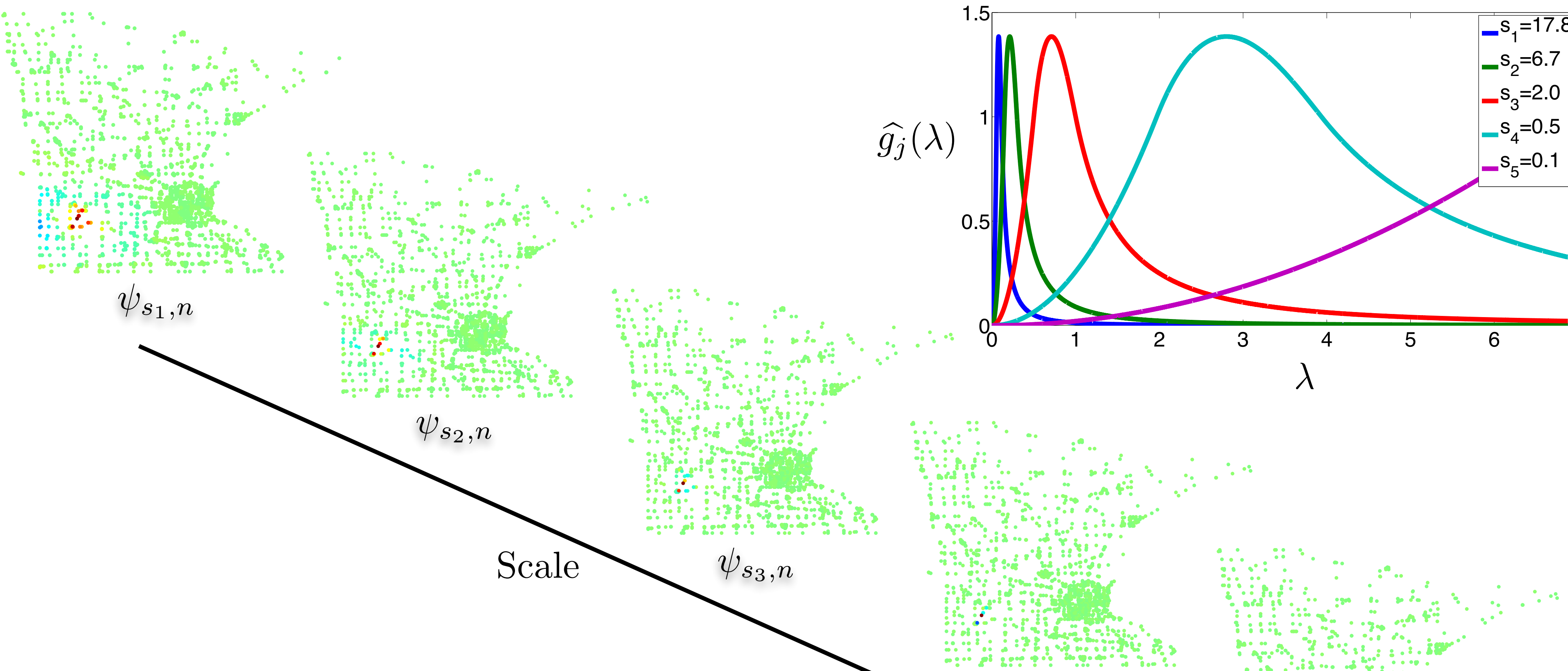
$$\frac{\psi_{t,j}(i)}{\|\psi_{t,j}\|} \leq Dt$$ for any t smaller than a critical scale function of $d_G(i,j)$

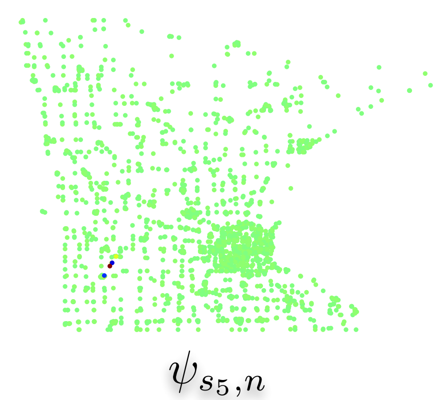
Reason ? At small scale, wavelet operator behaves like power of Laplacian

Spectral Graph Wavelet Localization



Spectral Graph Wavelet Localization

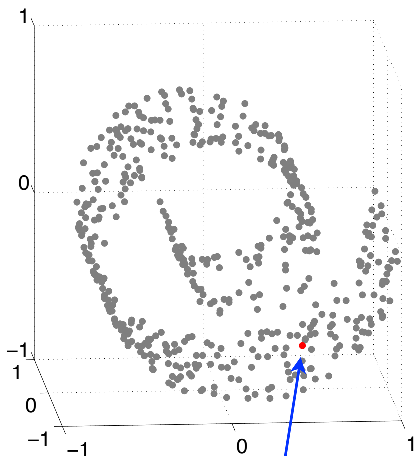




Characterizations of this localization

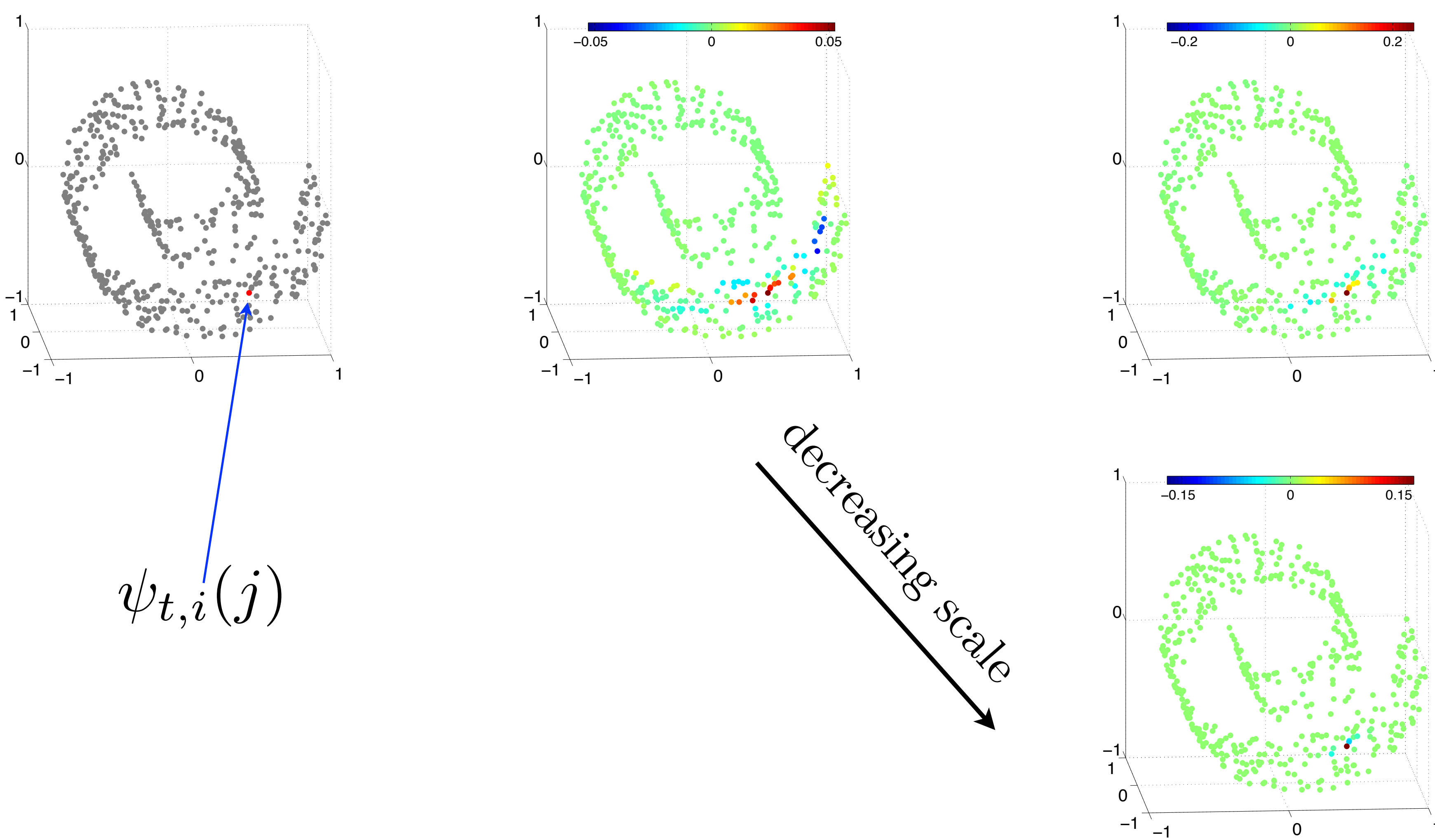
- Hammond et al., Wavelets on graphs via spectral graph theory, 2011
- Shuman et al., Vertex-frequency analysis on graphs, 2013

Scaling & Localization



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

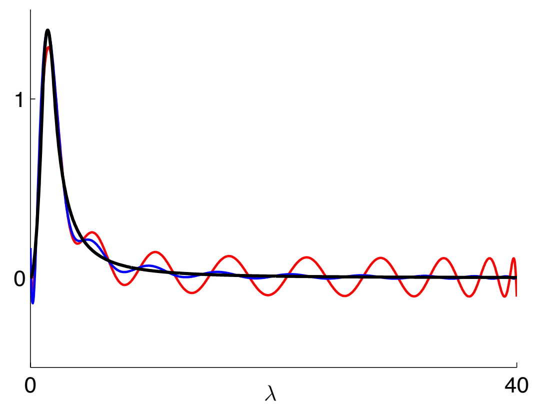
Scaling & Localization

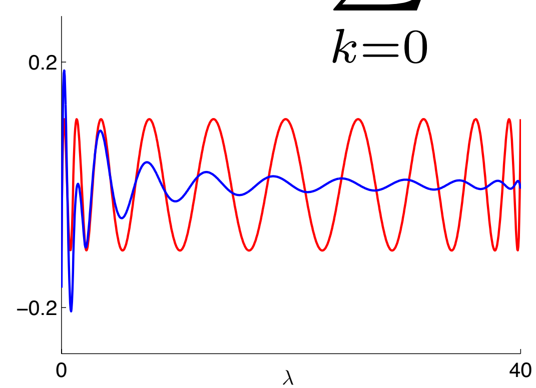


Remark on Implementation

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)

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

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

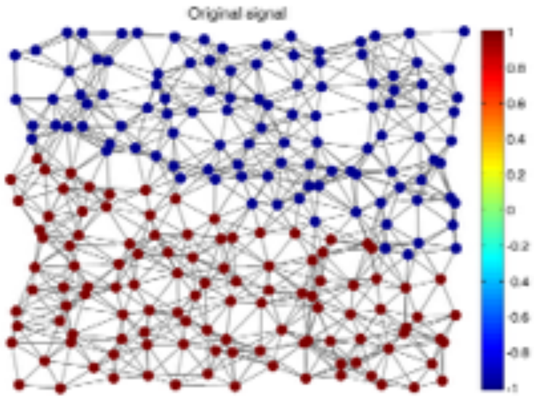
Shifted Chebyshev polynomial

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

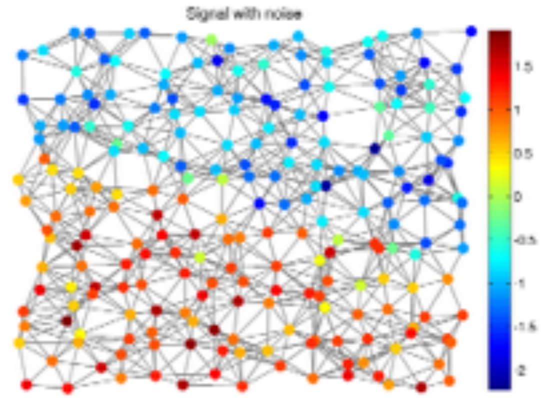
Complexity: $O(\sum_{n=1}^{J} M_n|E|)$ Note: “same” algorithm for adjoint !

Simple De-Noising with Wavelets

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



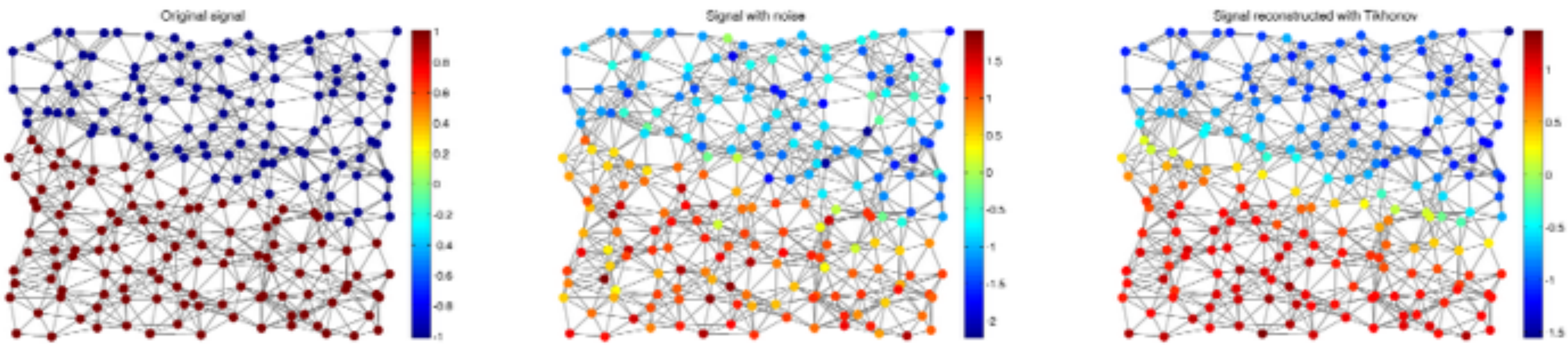
Original



Noisy

Simple De-Noising with Wavelets

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

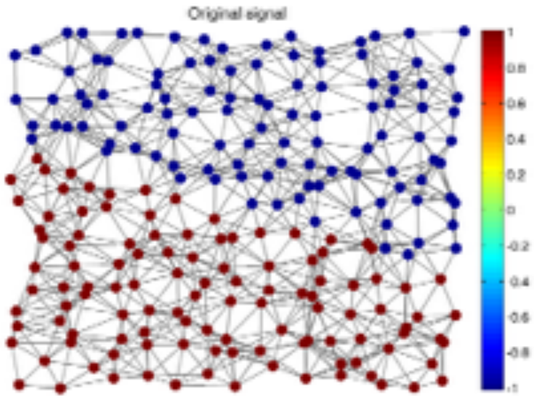


Original Noisy Denoised

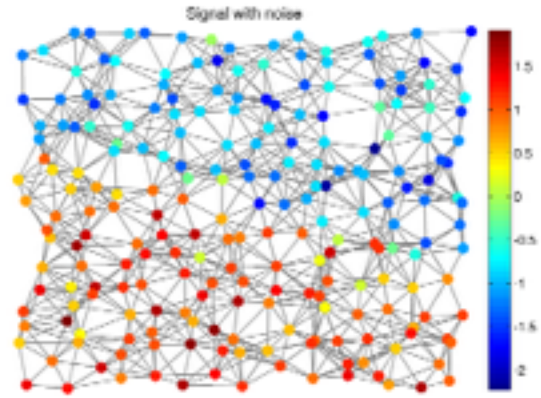
$$\text{argmin}_a \left\{ ||f - W^* a||_2^2 + \gamma ||a||_{1,\mu} \right\}$$

Simple De-Noising with Wavelets

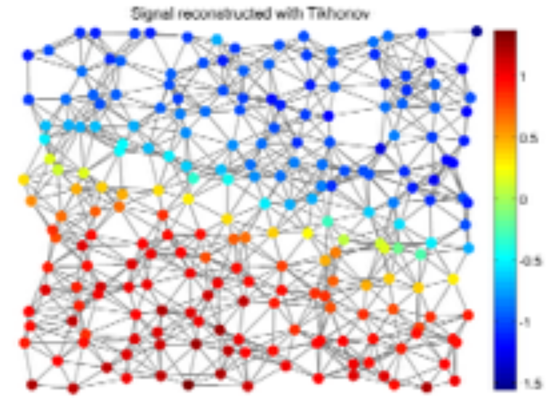
$$\operatorname{argmin}_f \left\{ ||f-y||_2^2 + \gamma f^T \mathcal{L} f \right\}$$



Original

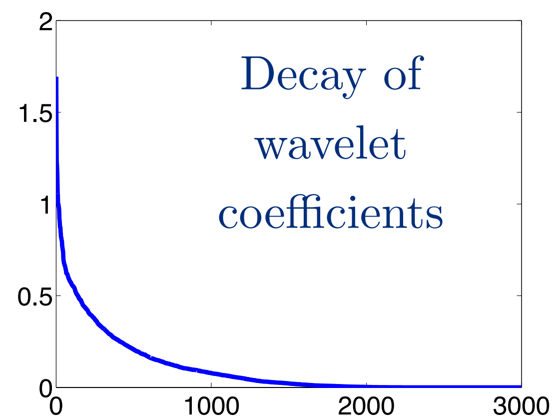


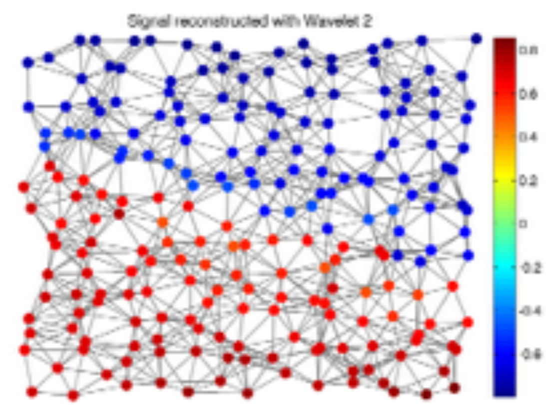
Noisy



Denoised

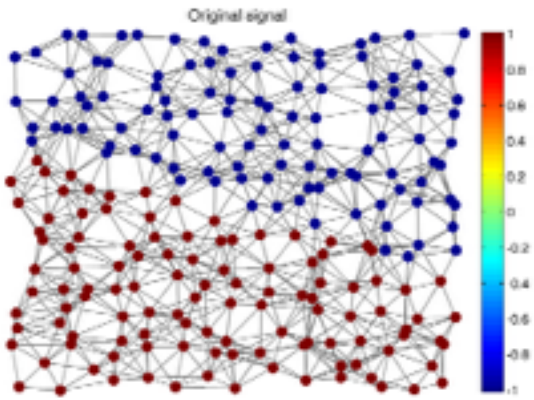
$$\operatorname{argmin}_a \left\{ ||f-W^* a||_2^2 + \gamma ||a||_{1,\nu} \right\}$$



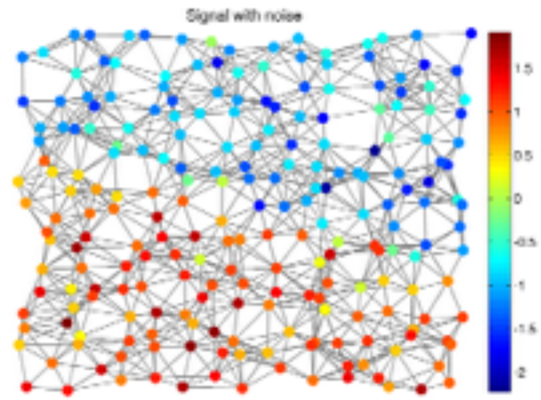


Simple De-Noising with Wavelets

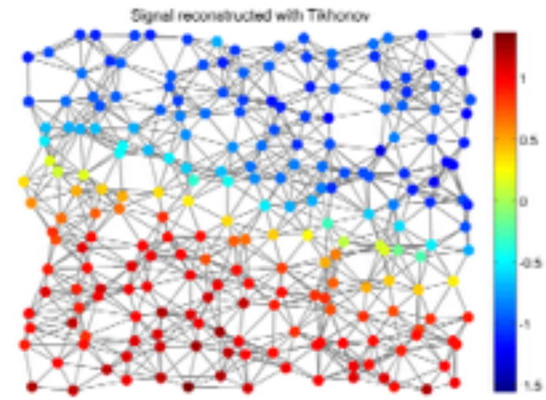
$$\operatorname{argmin}_f \{||f - y||_2^2 + \gamma f^T \mathcal{L} f\}$$



Original

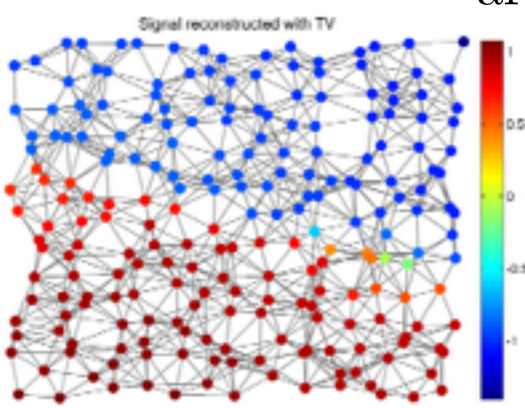


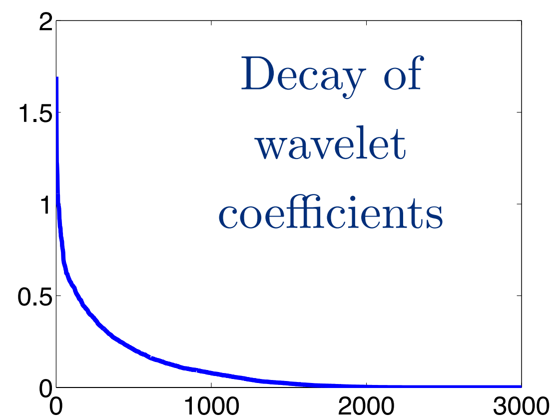
Noisy

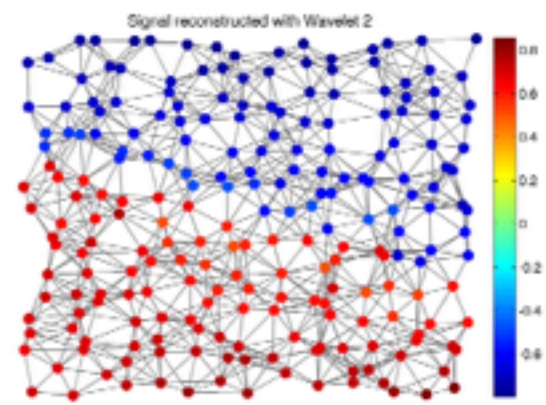


Denoised

$$\operatorname{argmin}_a \{||f - W^* a||_2^2 + \gamma ||a||_{1,\mu}\}$$







Transductive Learning

Let X be an array of data points $x_1, x_2, ..., x_n \in \mathbb{R}^d$

Each point has a desired class label $y_k \in Y$ (suppose binary)

At training you have the labels of a subset S of X $|S| = l < n$

Getting data is easy but labeled data is a scarce resource

GOAL: predict remaining labels

Rationale: minimize empirical risk on your training data such that

- your model is predictive
- your model is simple, does not overfit
- your model is “stable” (depends continuously on your training set)
- ...

Transductive Learning

Ex: Linear regression $y_k = \beta \cdot x_k + b$

Empirical Risk: $\|\mathbf{X}^t\beta - \mathbf{y}\|_2^2 \Longrightarrow \beta = (\mathbf{X}\mathbf{X}^t)^{-1}X\mathbf{y}$

if not enough observations, regularize (Tikhonov):

$$\|\mathbf{X}^t\beta - \mathbf{y}\|_2^2 + \alpha\|\beta\|_2^2 \Longrightarrow \beta = (\mathbf{X}\mathbf{X}^t + \alpha\mathbf{I})^{-1}X\mathbf{y}$$

Ridge Regression

Transductive Learning

Ex: Linear regression $y_k = \beta \cdot x_k + b$

Empirical Risk: $\|\mathbf{X}^t\beta - \mathbf{y}\|_2^2 \Longrightarrow \beta = (\mathbf{X}\mathbf{X}^t)^{-1}X\mathbf{y}$

if not enough observations, regularize (Tikhonov):

$$\|\mathbf{X}^t\beta - \mathbf{y}\|_2^2 + \alpha\|\beta\|_2^2 \Longrightarrow \beta = (\mathbf{X}\mathbf{X}^t + \alpha\mathbf{I})^{-1}X\mathbf{y}$$

Ridge Regression

Questions:

How can unlabelled data be used ?

More general linear model with a dictionary of features ?

$$\|\mathbf{\Phi}_X\beta - \mathbf{y}\|_{2,S}^2 + \alpha\mathcal{S}(\beta)$$

dictionary depends on data points

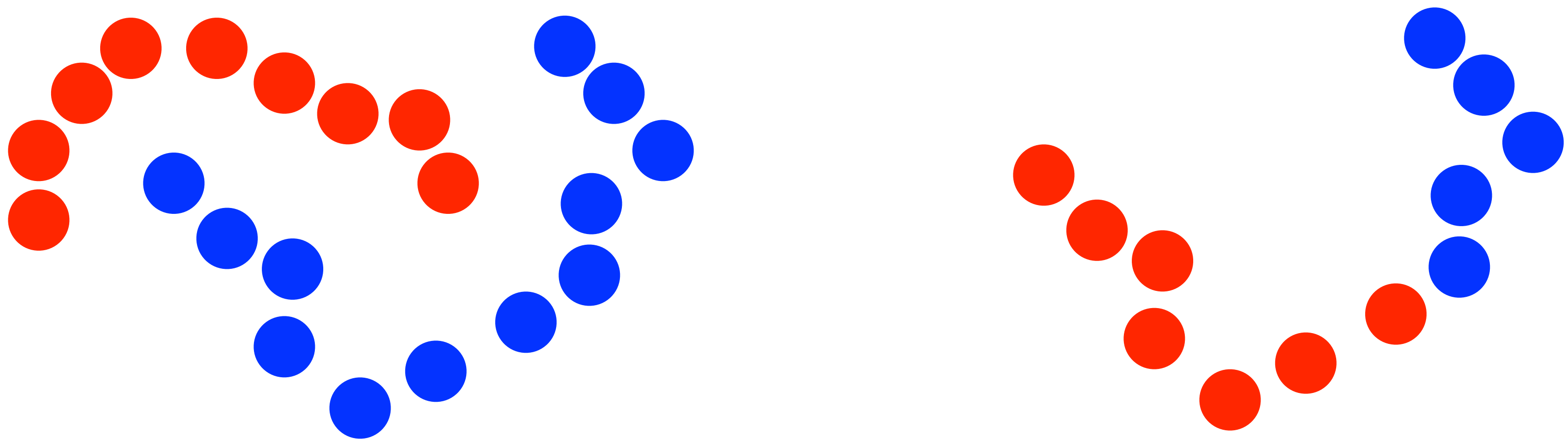
simplifies/stabilizes selected model

Learning on/with Graphs

How can unlabelled data be used ?

Assumption:

target function is not globally smooth but it is locally smooth over regions of data space that have some geometrical structure



Use graph to model this structure

Learning on/with Graphs

Example (Belkin, Niyogi)

Affinity between data points represented by edge weights (affinity matrix W)

measure of smoothness:
$$\Delta f = \sum_{i,j \in X} \mathbf{W}_{ij}(f(x_i) - f(x_j))^2$$
$$= \mathbf{f}^t L \mathbf{f} \quad L = W - D$$

Revisit ridge regression:
$$\|\mathbf{X}_S^t \beta - \mathbf{y}\|_2^2 + \alpha\|\beta\|_2^2 + \gamma\beta^t \mathbf{X} L \mathbf{X}^t \beta$$

Solution is smooth in graph "geometry"

Transduction & Representation

More general linear model with a dictionary of features ?

$\mathbf{\Phi}_X$ dictionary of features on the complete data set (data dependent)

$\boldsymbol{M}$ restricts to labeled data points (mask)

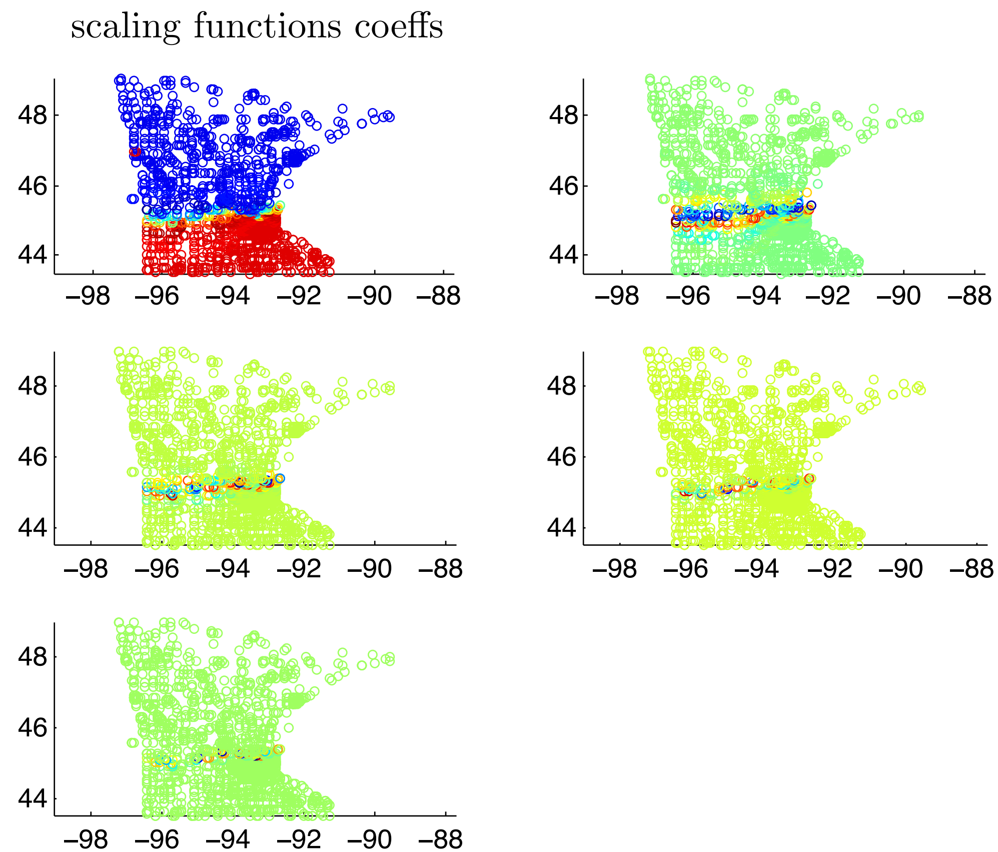
$$\arg\min_{\beta} \|\mathbf{y} - \mathbf{M}\mathbf{\Phi}_X \beta\|_2^2 + \alpha \mathcal{S}(\beta)$$

Empirical Risk

Model Selection penalty, sparsity ?
Smoothness on graph ?

<u>Important Note:</u> our dictionary will be data dependent but its construction is not part of the above optimization

Sparsity and Smoothness on Graphs



Sparsity and Transduction

$$\arg\min_{\beta} \|\mathbf{y} - \mathbf{M}\mathbf{\Phi}_X \beta\|_2^2 + \alpha \boxed{\mathcal{S}(\beta)}$$

Since sparsity = smoothness on graph, why not simple LASSO ?

$$\arg\min_{\beta} \|\mathbf{y} - \mathbf{M}\mathbf{\Phi}_X \beta\|_2^2 + \alpha \|\beta\|_1$$

Sparsity and Transduction

$$\arg\min_{\beta} \|\mathbf{y} - \mathbf{M}\mathbf{\Phi}_X \beta\|_2^2 + \alpha \mathcal{S}(\beta)$$

Since sparsity = smoothness on graph, why not simple LASSO ?

$$\arg\min_{\beta} \|\mathbf{y} - \mathbf{M}\mathbf{\Phi}_X \beta\|_2^2 + \alpha \|\beta\|_1$$

Bad Idea:

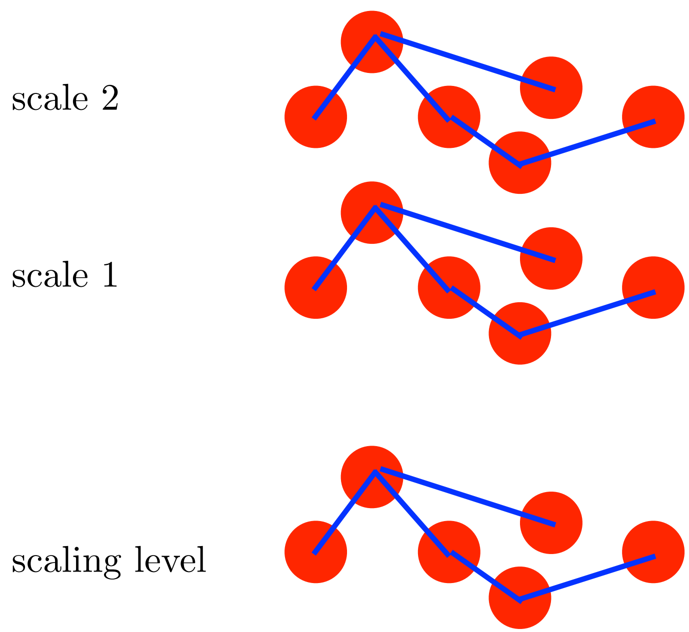
We *know* there are strongly correlated coefficients (LASSO will kill some of them)

There is no information to determine masked wavelets

Group Sparsity - take I

Scaling functions not sparse are optimized separately

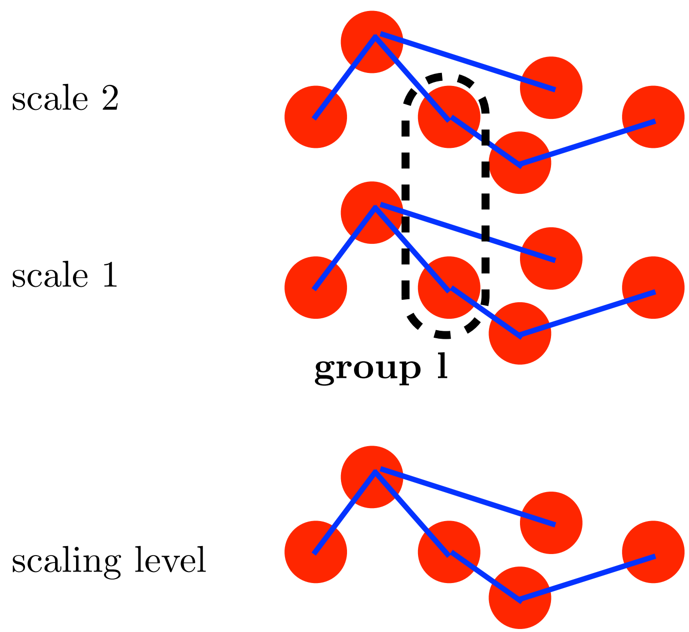
Group potentially correlated variables (scales)



Group Sparsity - take I

Scaling functions not sparse are optimized separately

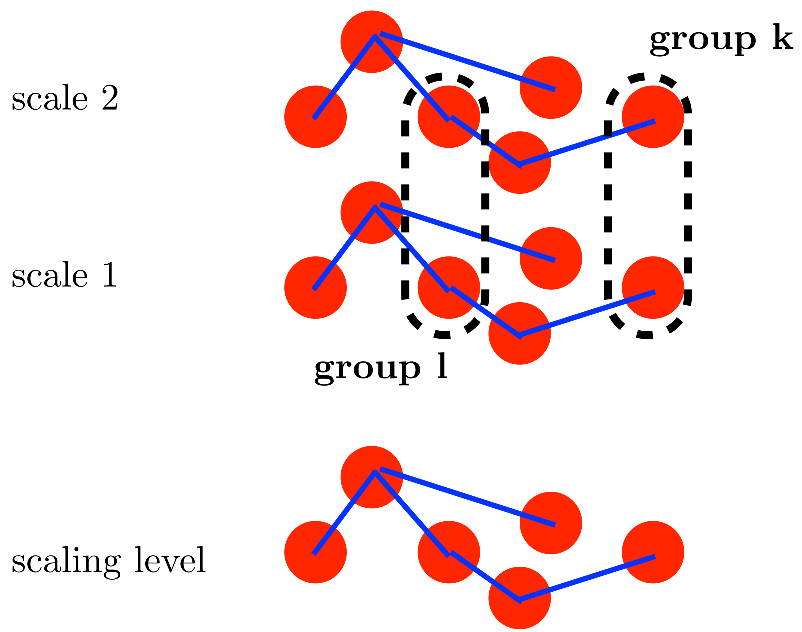
Group potentially correlated variables (scales)



Group Sparsity - take I

Scaling functions not sparse are optimized separately

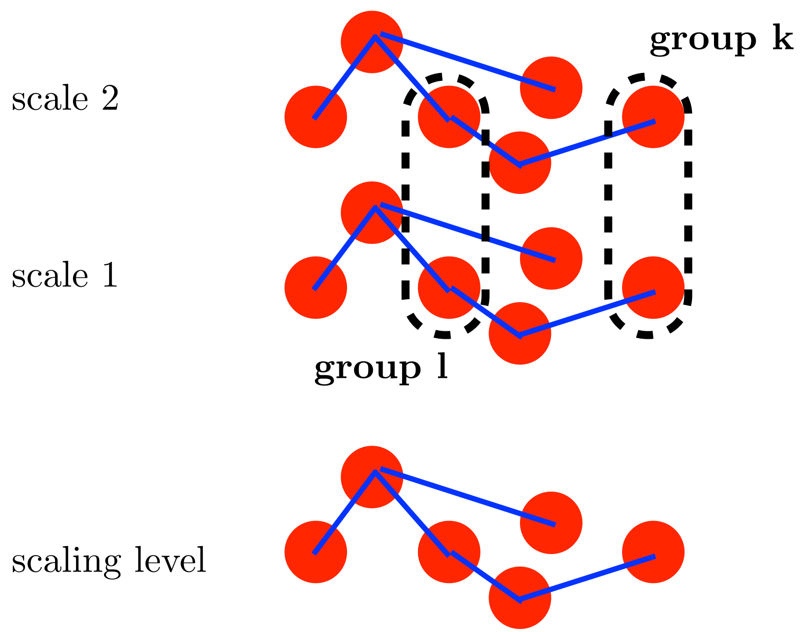
Group potentially correlated variables (scales)



Group Sparsity - take I

Scaling functions not sparse are optimized separately

Group potentially correlated variables (scales)



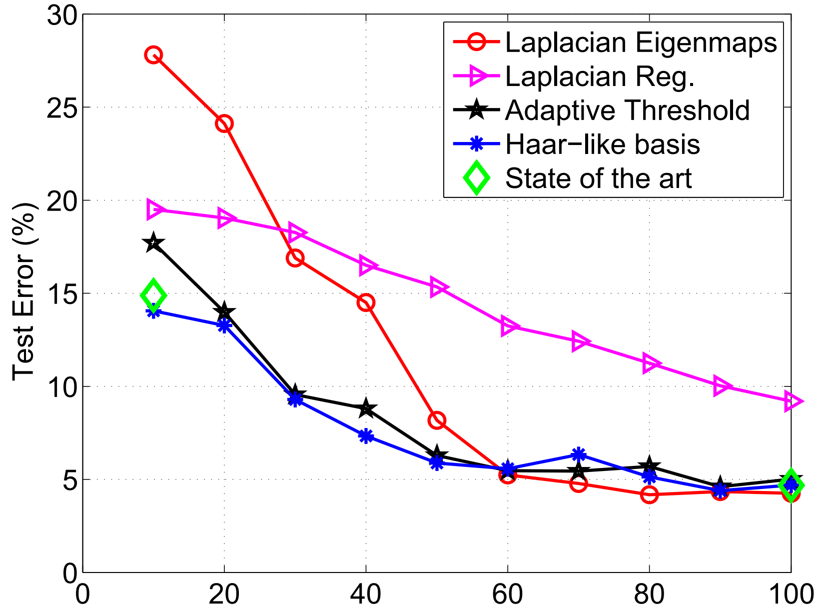
Few groups should be active = local smoothness

Inside group, all coefficients can be active

Formulate with mixed-norms $\|\beta\|_{p,q}$

Simple model, no overlap, optimized like LASSO

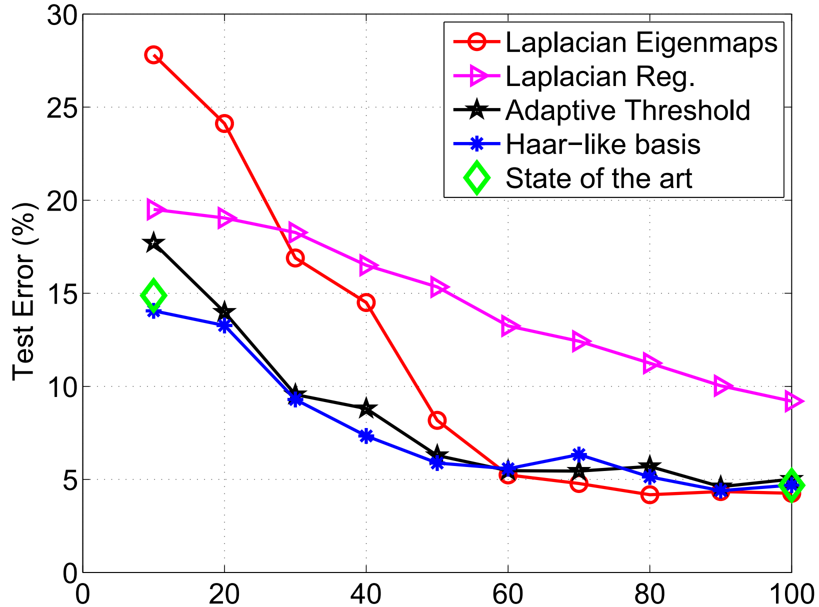
Preliminary Results



2-class USPS

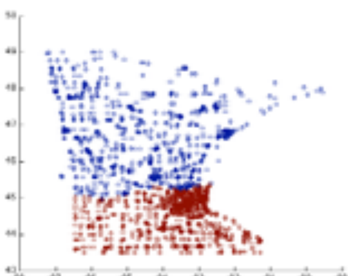
Simulation results from Gavish et al, ICML 2010

Preliminary Results

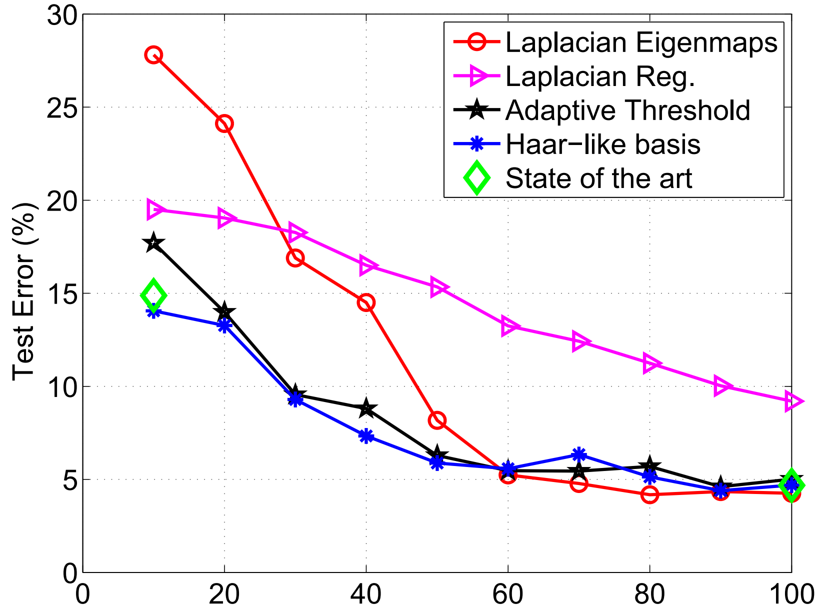


2-class USPS

Simulation results from Gavish et al, ICML 2010



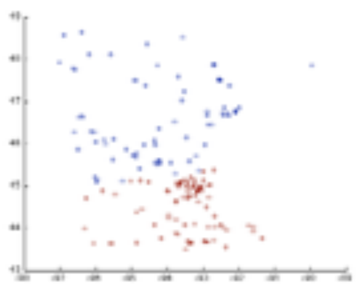
Preliminary Results



2-class USPS

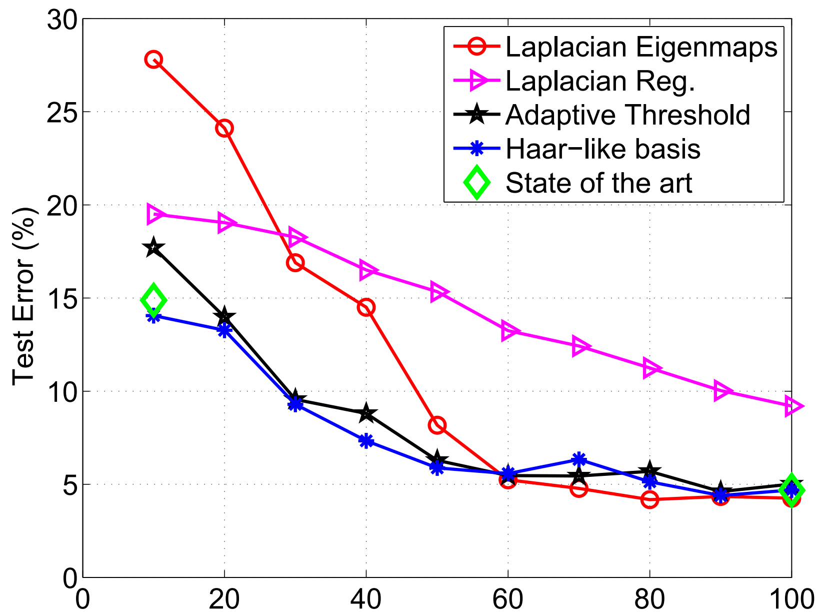
Simulation results from Gavish et al, ICML 2010

5% labeled



recovered

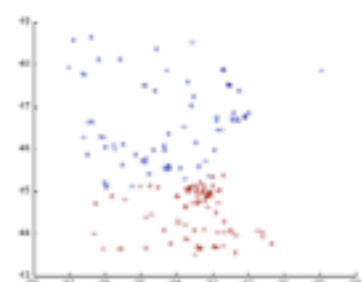
Preliminary Results



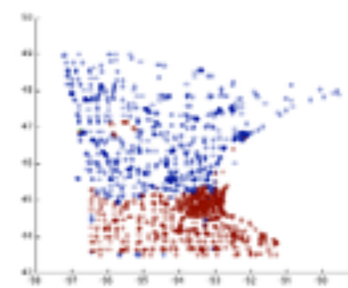
2-class USPS

Simulation results from Gavish et al, ICML 2010

5% labeled



recovered



Is it spectacular ?

No. Comparable to state-of-art :(

Example: Shape Descriptors

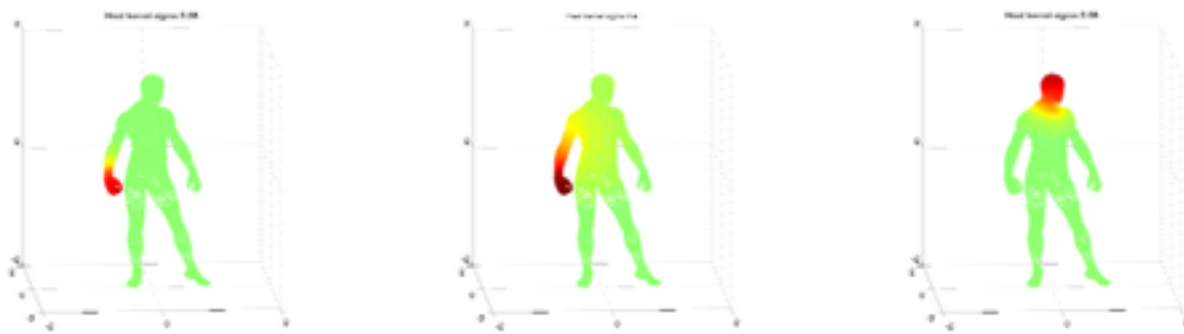
Shape represented by 3D point cloud

Construct graph

k-Nearest Neighbors

ϵ – Neighborhood

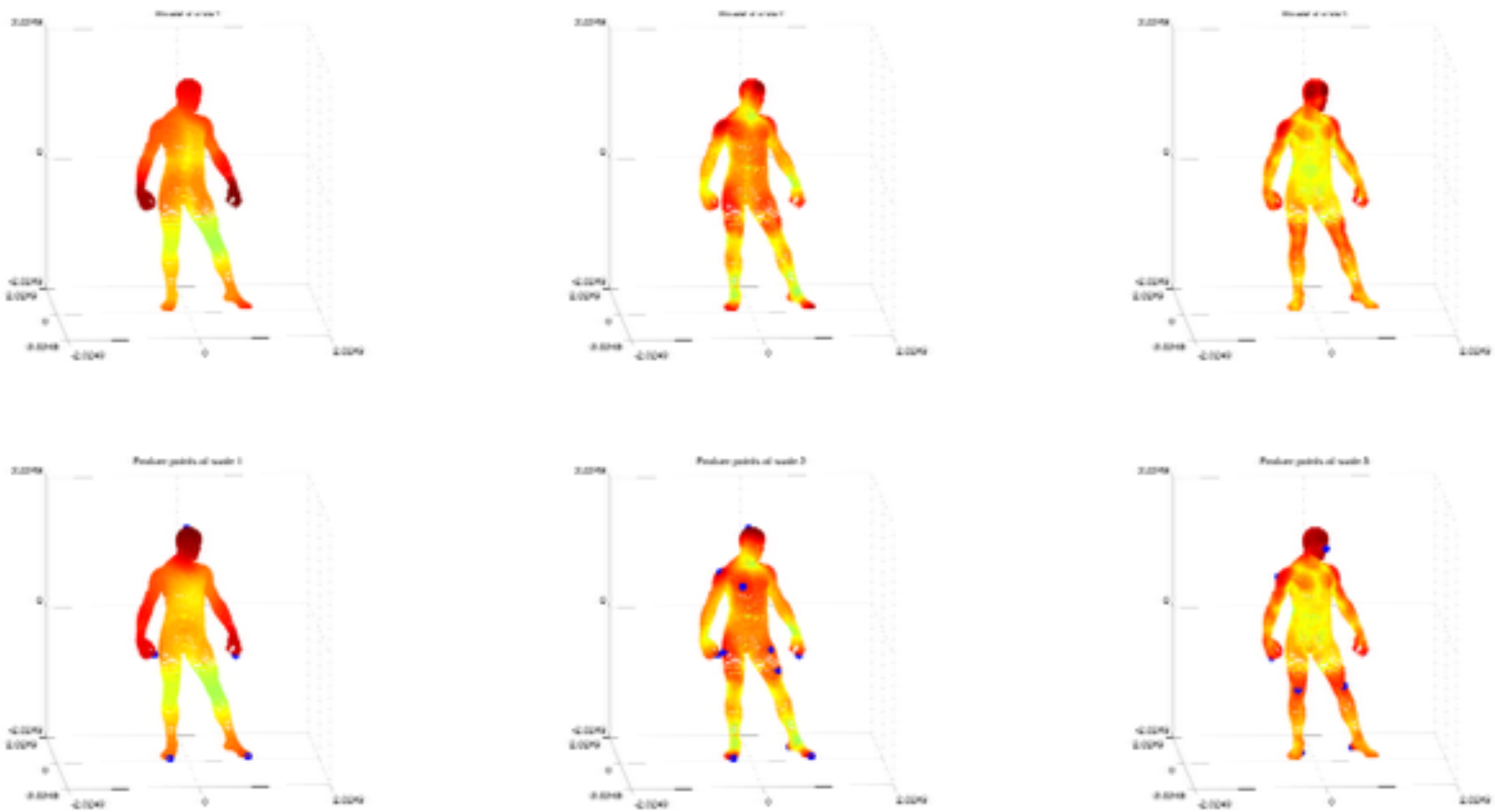
Ex: Localized heat kernel on point clouds



Example: Shape Descriptors

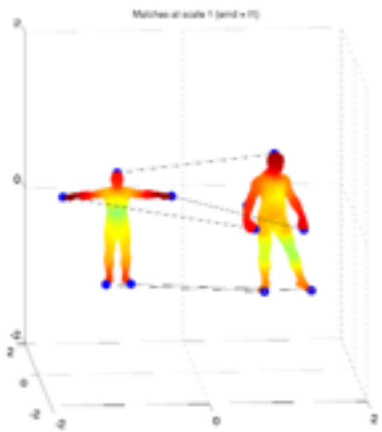
Idea: use multiscale localized features on graph

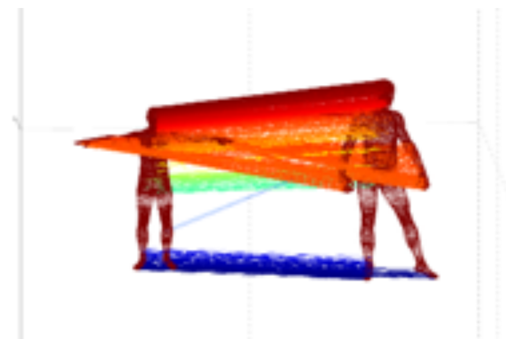
Ex: graph wavelet transform of coordinates maps



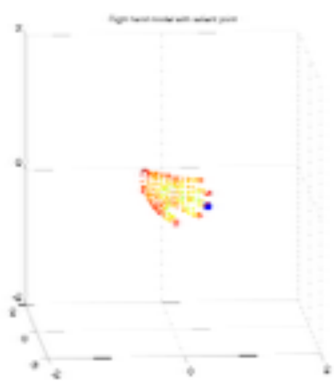
Example: Shape Descriptors

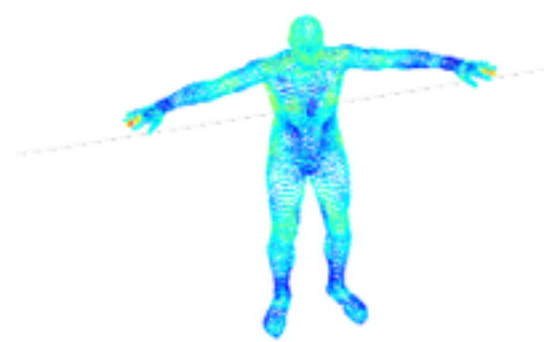
Application 1: sparse/dense description & robust matching





Application 2: parts matching





Outline

- Introduction:
  - Graphs and elements of spectral graph theory
- Kernel Convolution:
  - Localization, filtering, smoothing and applications
- Spectral Graph Wavelets
- Multiresolution
- From Graphs to Manifolds

Outline

- Introduction:
  - Graphs and elements of spectral graph theory
- Kernel Convolution:
  - Localization, filtering, smoothing and applications
- Spectral Graph Wavelets
- Multiresolution
- ~~From Graphs to Manifolds~~

Graph wavelets

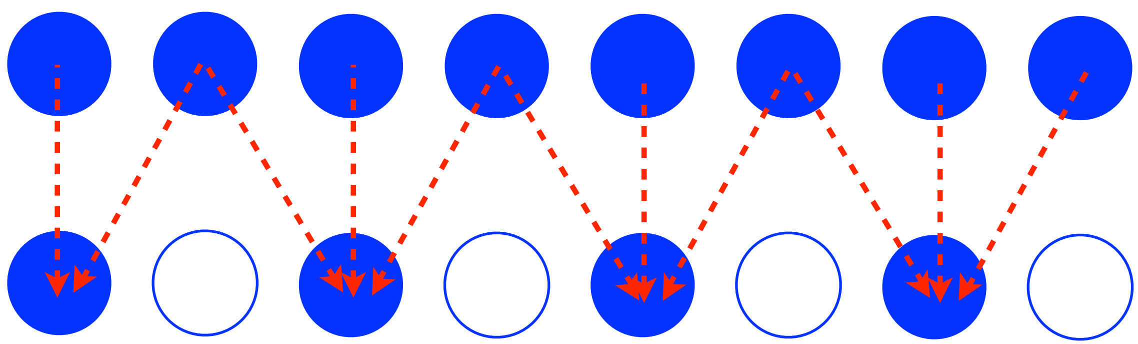
- Redundancy versus sparsity
  - can we remove some or all of it ?
- Faster algorithms
  - traditional wavelets have fast filter banks implementation
  - whatever scale, you use the same filters
  - here: large scales -> more computations
- Goal: solve both problems at one

Basic Ingredients

Euclidean multiresolution is based on two main operations

Filtering (typically low-pass and high-pass)

Down and Up sampling

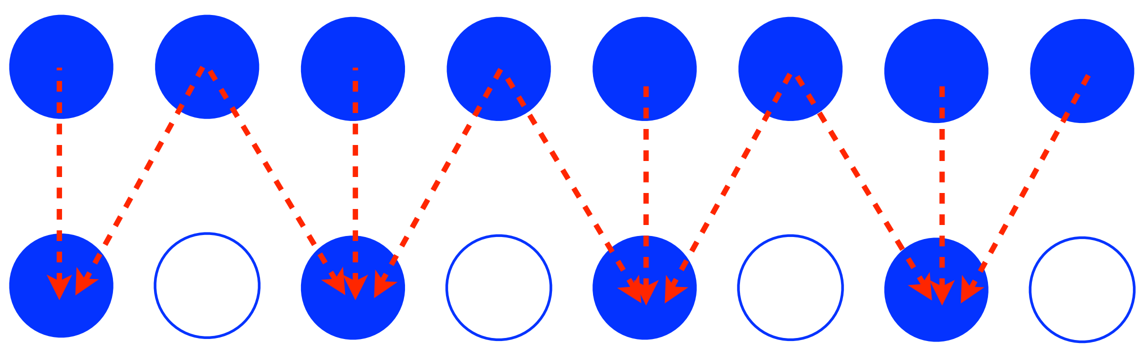


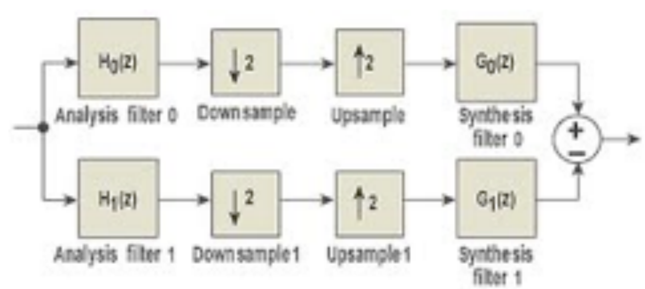
Basic Ingredients

Euclidean multiresolution is based on two main operations

Filtering (typically low-pass and high-pass)

Down and Up sampling



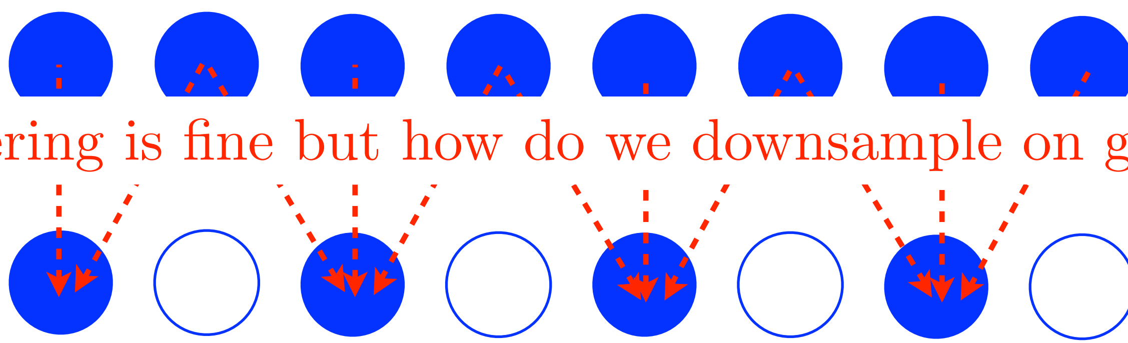


Basic Ingredients

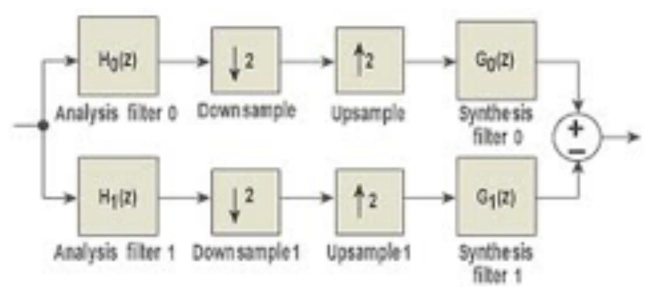
Euclidean multiresolution is based on two main operations

Filtering (typically low-pass and high-pass)

Down and Up sampling

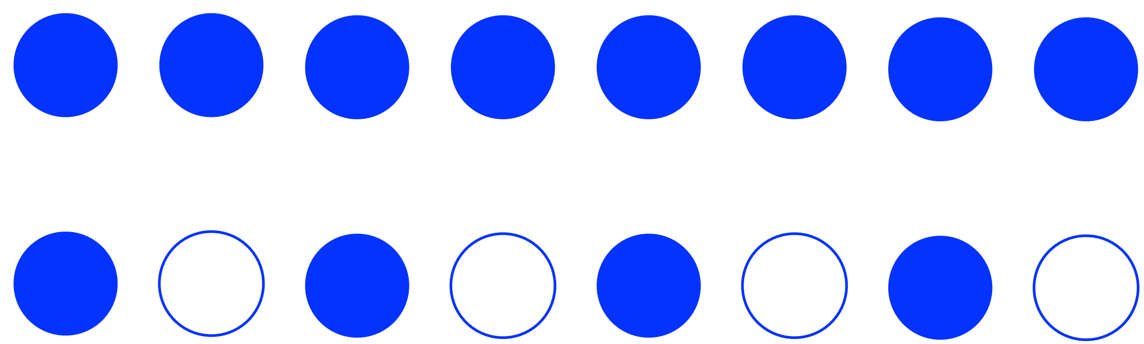


Filtering is fine but how do we downsample on graphs ???



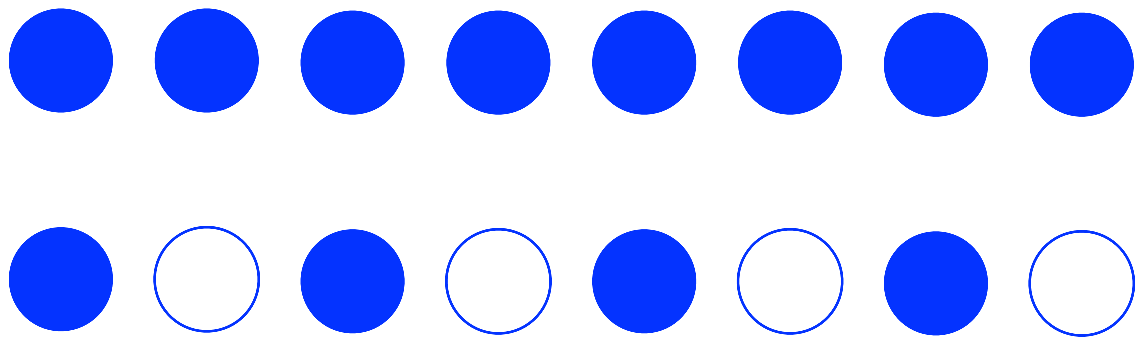
Basic Ingredients

Subsampling is equivalent to splitting in two cosets (even, odd)



Basic Ingredients

Subsampling is equivalent to splitting in two cosets (even, odd)



Questions: How do we partition a graph into meaningful cosets ?

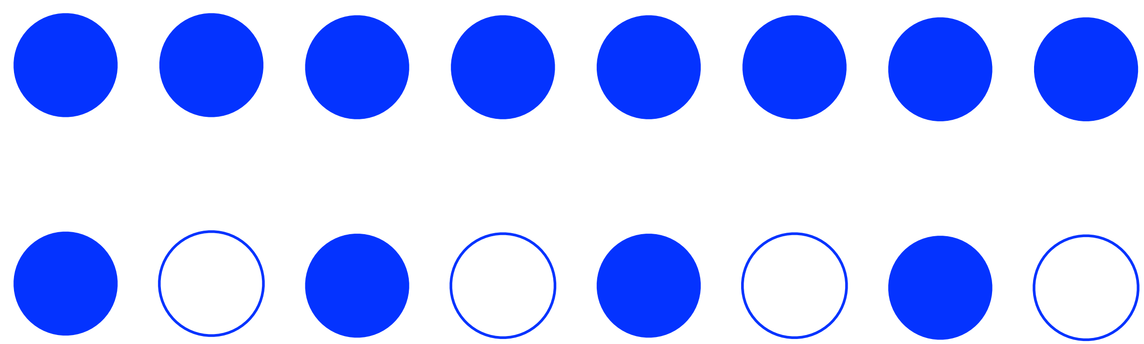
Are there efficient algorithms for these partitions ?

Are there theoretical guarantees ?

How do we define a new graph from the cosets ?

Cosets - A spectral view

Subsampling is equivalent to splitting in two cosets (even, odd)



Classically, selecting a coset can be interpreted easily in Fourier:

$$f_{\text{sub}}(i) = \frac{1}{2} f(i) \big(1 + \cos(\pi i)\big)$$

eigenvector of largest eigenvalue

Cosets and Nodal Domains

Nodal domain: maximally connected subgraph s.t. all vertices have same sign w.r.t a reference function

We would like to find a very large number of nodal domains, ideally $|V|$!

Nodal domains of Laplacian eigenvectors are special (and well studied)

Cosets and Nodal Domains

Nodal domain: maximally connected subgraph s.t. all vertices have same sign w.r.t a reference function

We would like to find a very large number of nodal domains, ideally $|V|$!

Nodal domains of Laplacian eigenvectors are special (and well studied)

Theorem: the number of nodal domains associated to the largest laplacian eigenvector of a connected graph is maximal,

$$\nu(\phi_{\max}) = \nu(G) = |V|$$

IFF G is bipartite

In general: $\nu(G) = |V| - \chi(G) + 2$ (extreme cases: bipartite and complete graphs)

Cosets and Nodal Domains

Nodal domain: maximally connected subgraph s.t. all vertices have same sign w.r.t a reference function

We would like to find a very large number of nodal domains, ideally $|V|$!

Nodal domains of Laplacian eigenvectors are special (and well studied)

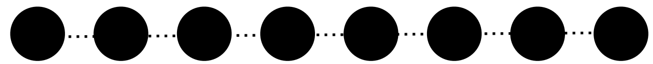
For any connected graph we will thus naturally define cosets and their associated selection functions

$$V_+ = \{i \in V \text{ s.t. } \phi_{N-1}(i) \geq 0\} \qquad V_- = \{i \in V \text{ s.t. } \phi_{N-1}(i) < 0\}$$

$$M_+(i) = \frac{1}{2}\big(1 + \mathrm{sgn}(\phi_{N-1}(i))\big) \qquad M_-(i) = \frac{1}{2}\big(1 - \mathrm{sgn}(\phi_{N-1}(i))\big)$$

Examples of cosets

Simple line graph



$$\phi_k(u) = \sin(\pi k u/n + \pi/2n) \qquad \lambda_k = 2 - 2\cos(\pi k/n) \qquad 1 \leq k \leq n$$

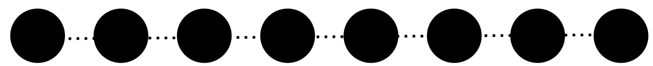
Examples of cosets

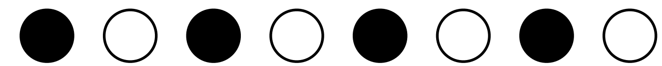
Simple line graph

$$\phi_k(u) = \sin(\pi k u/n + \pi/2n) \qquad \lambda_k = 2 - 2\cos(\pi k/n) \qquad 1 \leq k \leq n$$

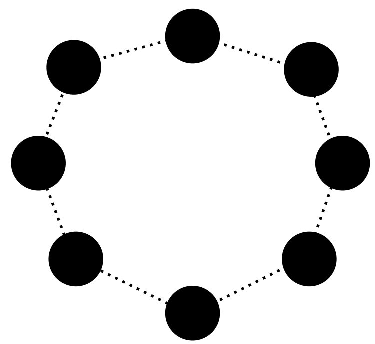
Examples of cosets

Simple line graph





Simple ring graph

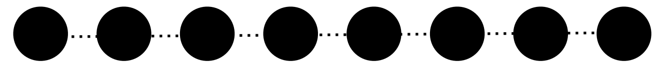


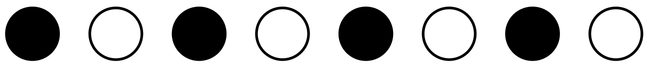
$$\phi_k^1(u) = \sin(2\pi k u/n) \qquad \phi_k^2(u) = \cos(2\pi k u/n) \quad 1 \leq k \leq n/2$$

$$\lambda_k = 2 - 2\cos(2\pi k/n)$$

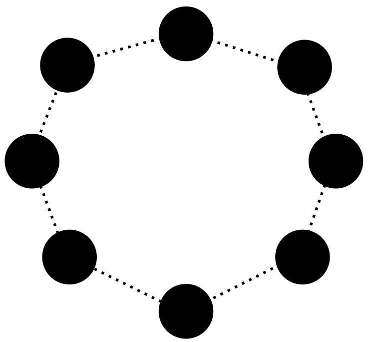
Examples of cosets

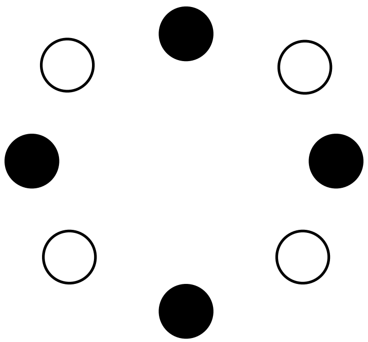
Simple line graph





Simple ring graph





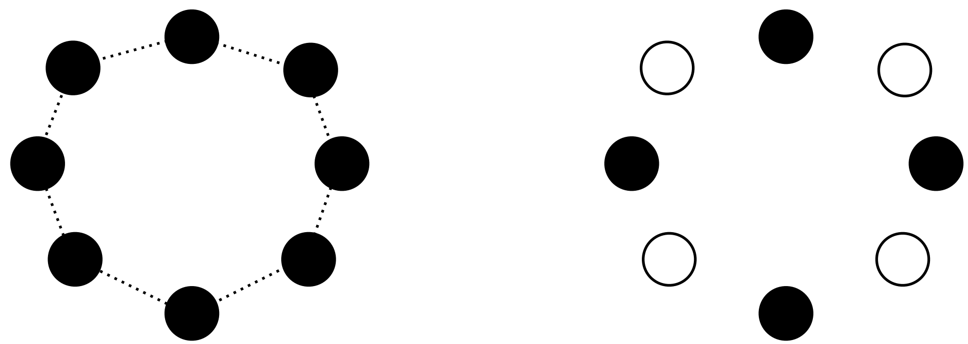
$$\phi_k^1(u) = \sin(2\pi k u/n) \qquad \phi_k^2(u) = \cos(2\pi k u/n) \quad 1 \leq k \leq n/2$$

$$\lambda_k = 2 - 2\cos(2\pi k/n)$$

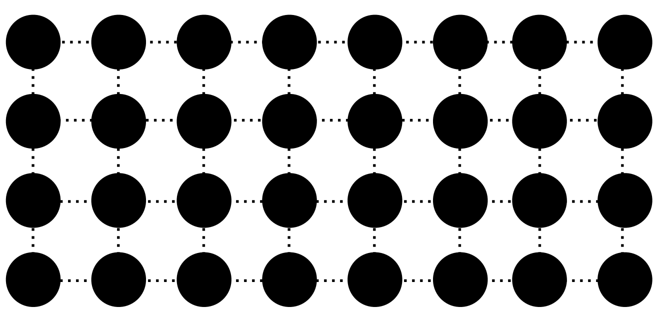
Examples of cosets

Simple line graph

Simple ring graph



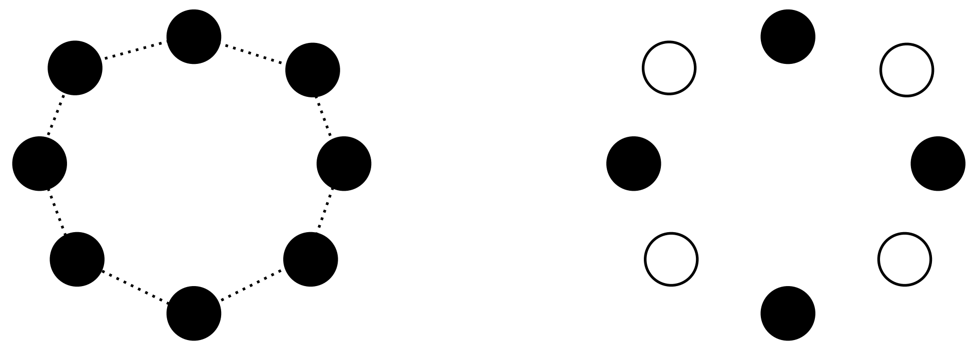
Lattice



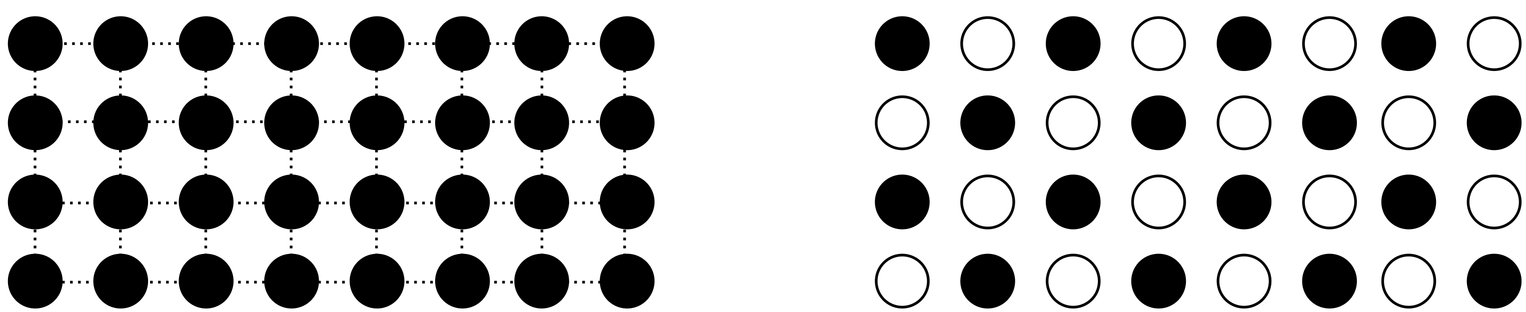
Examples of cosets

Simple line graph

Simple ring graph



Lattice



quincunx

The Agonizing Limits of Intuition

- Multiplicity of $\lambda_{\max}$
  - how do we choose the control vector in that subspace ?
  - even a prescription can be numerically ill-defined
  - graphs with "flat" spectrum in close to their spectral radius
- Laplacian eigenvectors do not always behave like global oscillations
  - seems to be true for random perturbations of simple graphs
  - true even for a class of trees [Saito2011]

A Laplacian Pyramid on Graphs

A Laplacian Pyramid on Graphs

Single level pyramid

Filtering

Downsampling

A Laplacian Pyramid on Graphs

Single level pyramid

Filtering

Downsampling

Graph reduction

Coarsening

Prediction/Interpolation

A Laplacian Pyramid on Graphs

Single level pyramid

Filtering

Downsampling

Graph reduction

Coarsening

Prediction/Interpolation

Graph sparsification

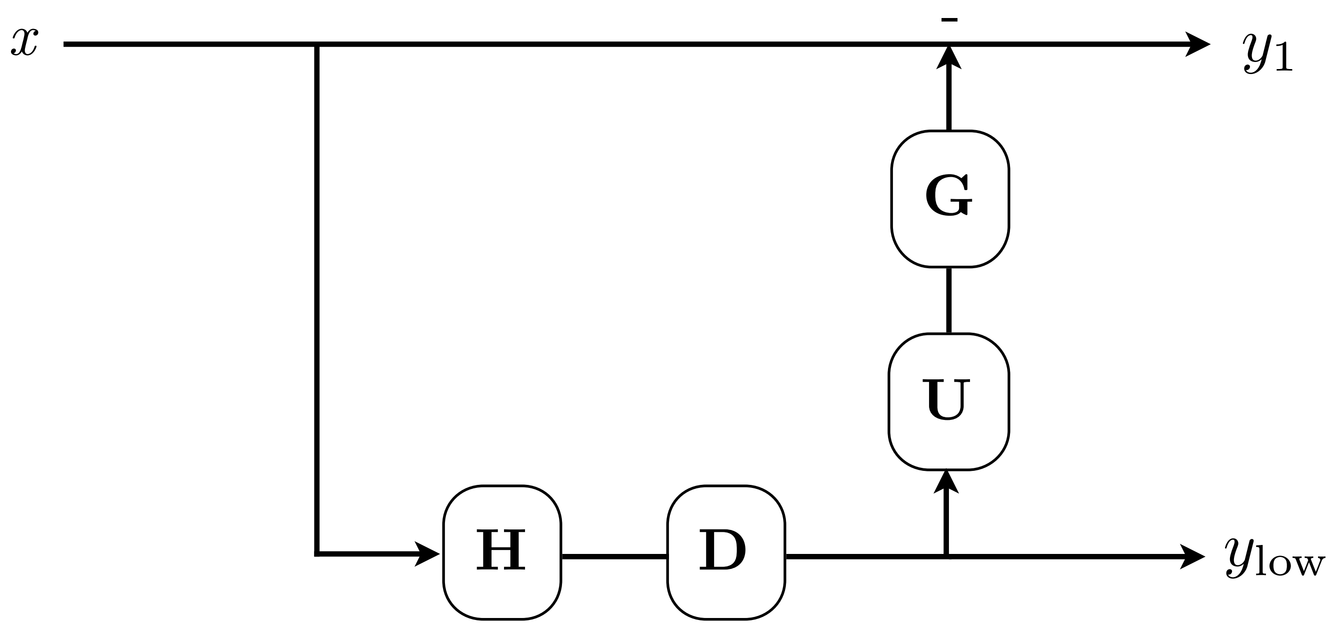
A Laplacian Pyramid on Graphs

Single level pyramid	Filtering Downsampling
Graph reduction	Coarsening Prediction/Interpolation
Graph sparsification	
Single level pyramid	

Shuman, Faraji, VDG, A framework for multiscale transforms on graphs, 2013

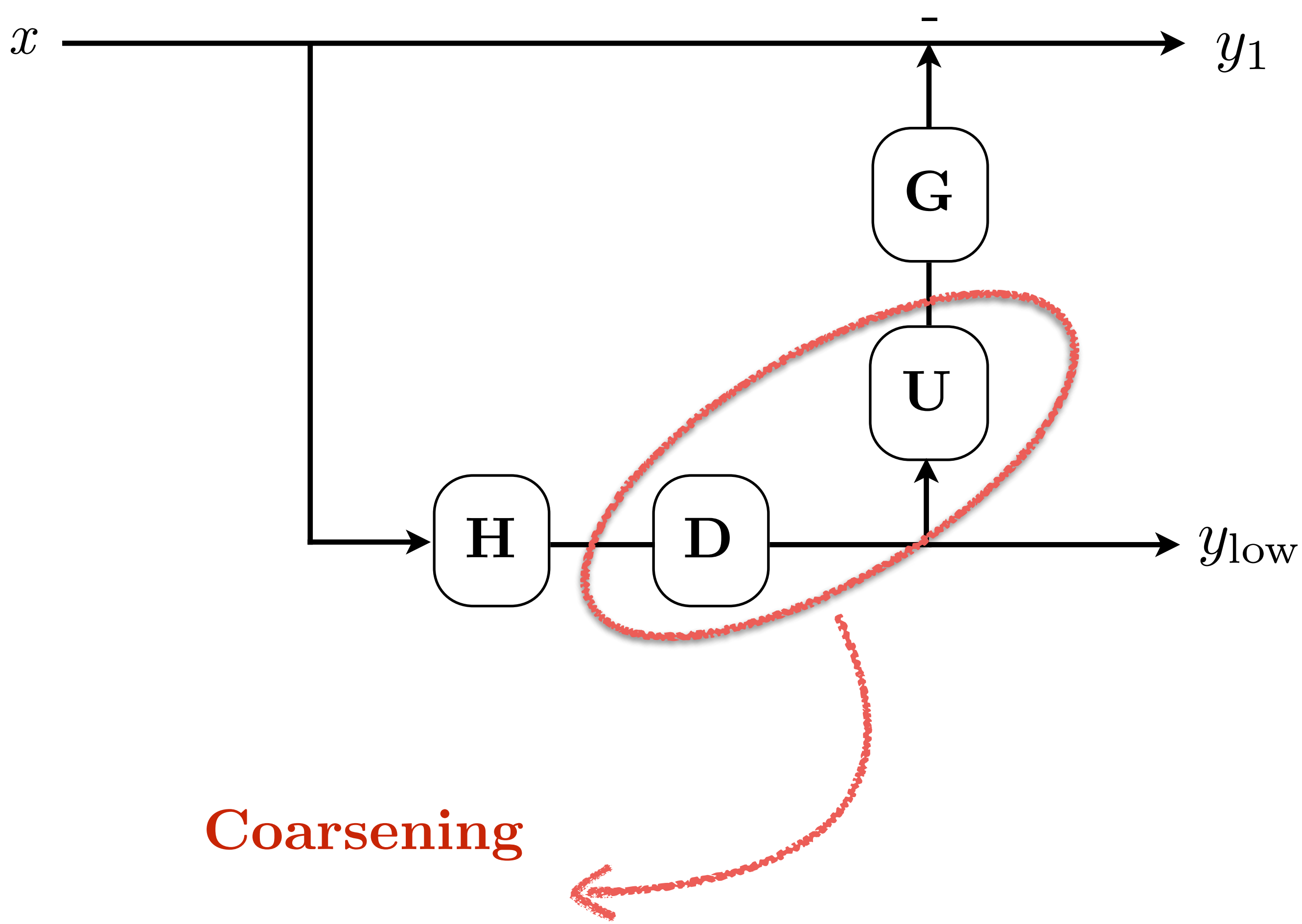
The Laplacian Pyramid

Analysis operator



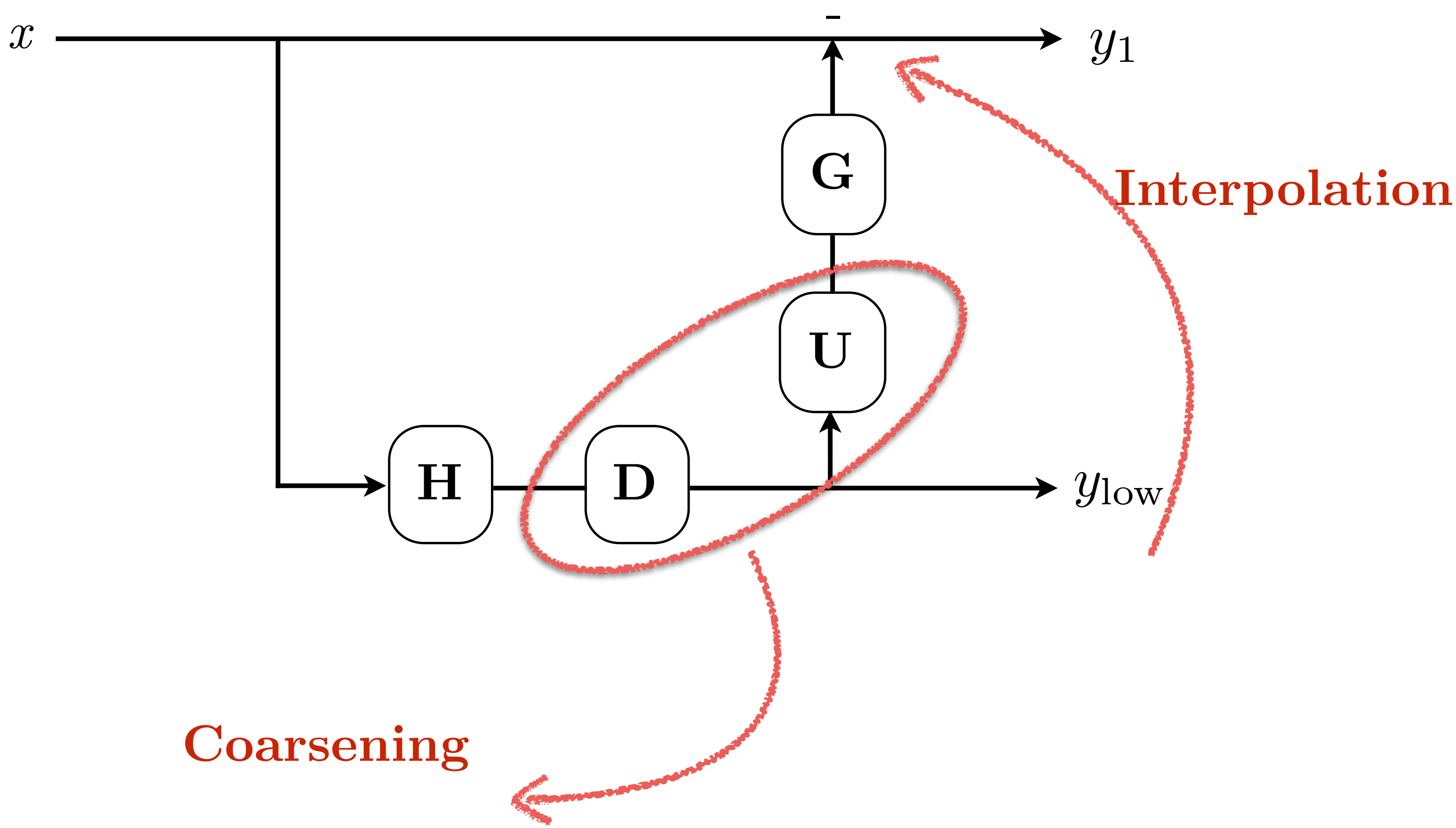
The Laplacian Pyramid

Analysis operator



The Laplacian Pyramid

Analysis operator



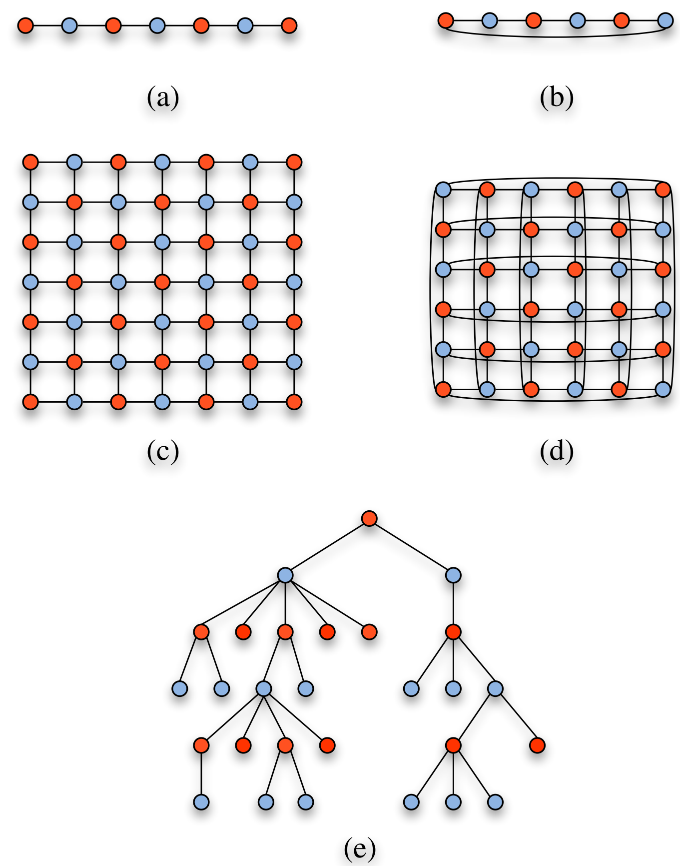
Downsampling

$$\mathcal{V}_1 = \mathcal{V}_+ := \{i \in \mathcal{V} : u_{\max}(i) \geq 0\}$$

Relaxed solution to 2-coloring for regular graphs

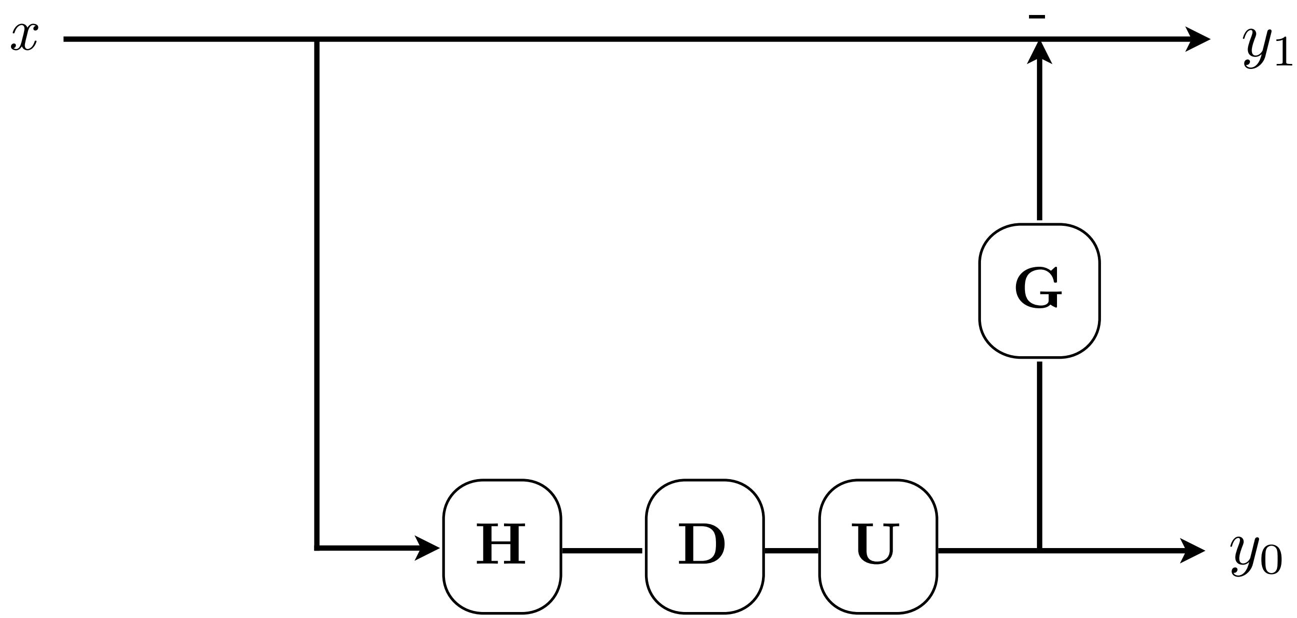
Exact for bipartite graphs

Connections with nodal domains theory for laplacian eigenvectors



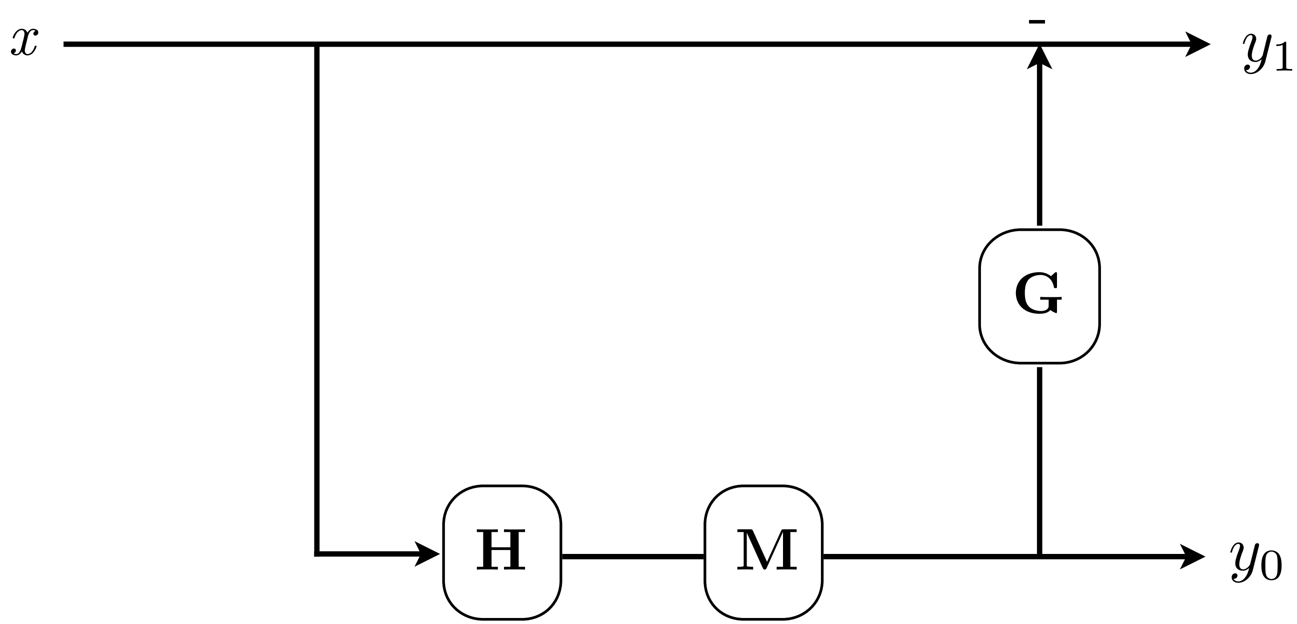
The Laplacian Pyramid

Analysis operator



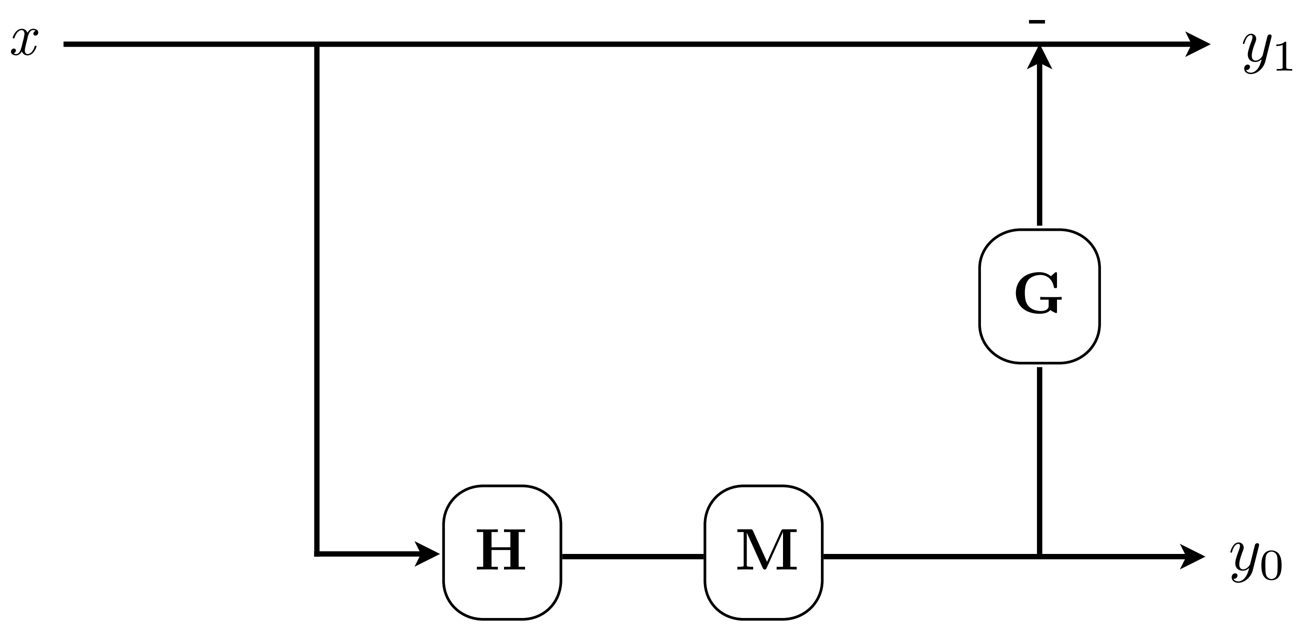
The Laplacian Pyramid

Analysis operator



The Laplacian Pyramid

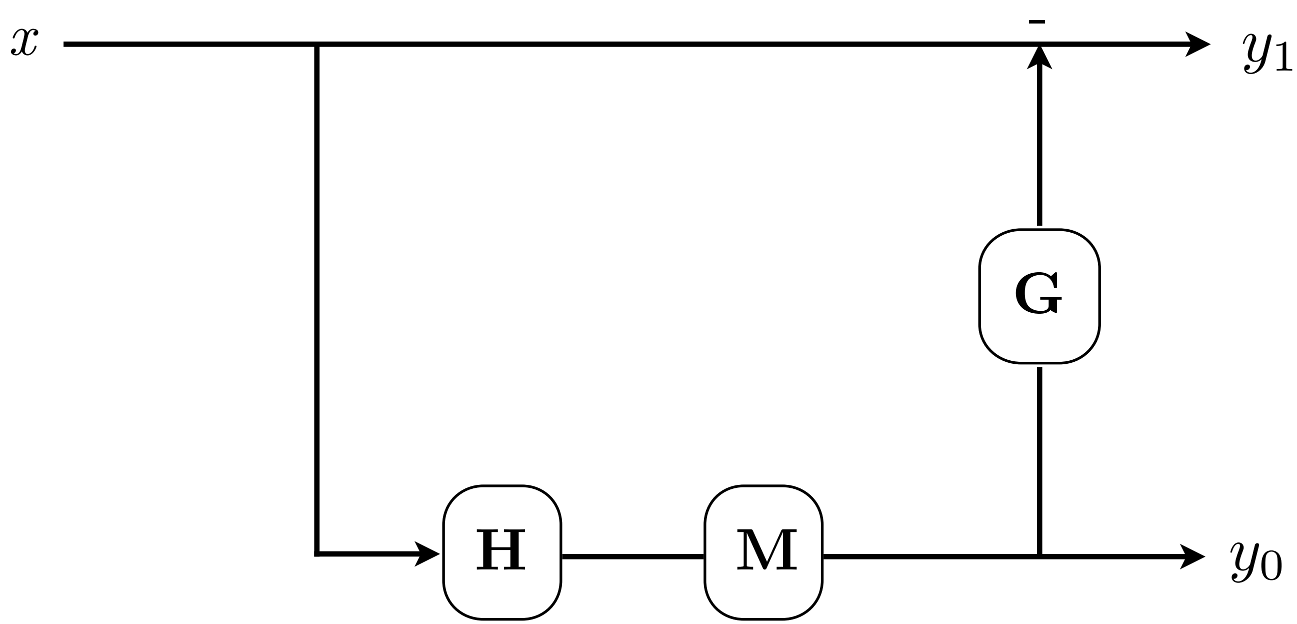
Analysis operator



$$y_0 = \mathbf{H_m}x = \mathbf{MH}x \qquad y_1 = x - \mathbf{G}y_0 = x - \mathbf{GH_m}x$$

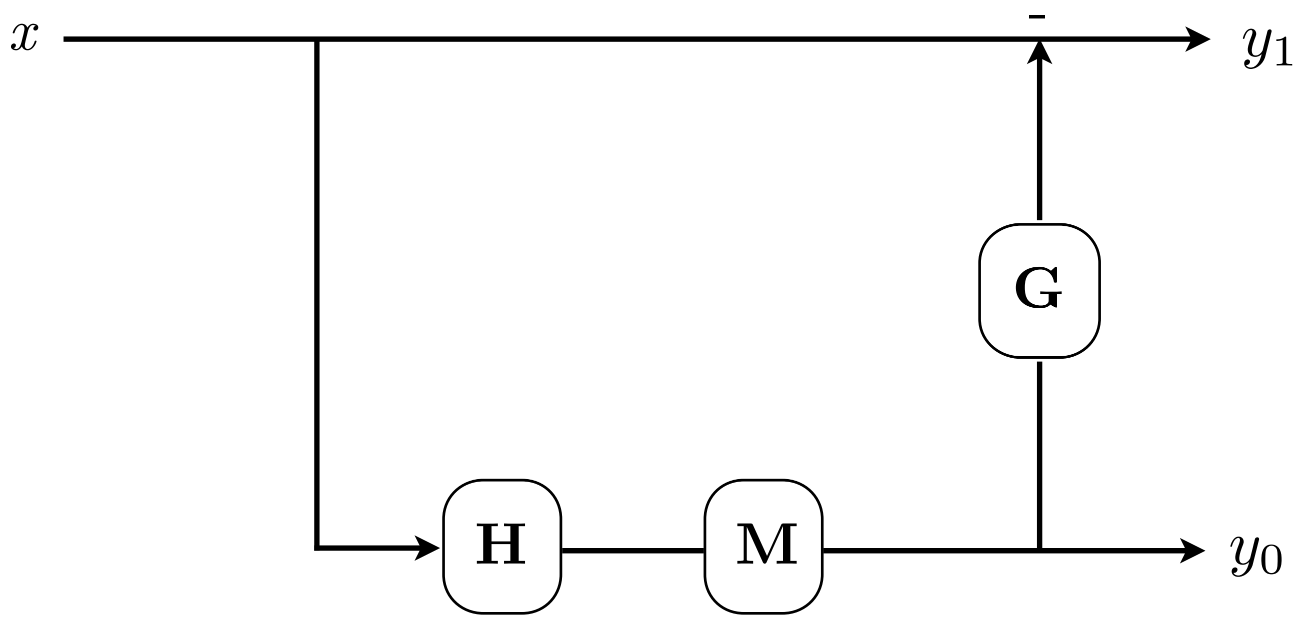
The Laplacian Pyramid

Analysis operator



The Laplacian Pyramid

Analysis operator



$$\underbrace{\begin{pmatrix} y_0 \\ y_1 \end{pmatrix}}_{y} = \underbrace{\begin{pmatrix} \mathbf{H_m} \\ \mathbf{I} - \mathbf{G}\mathbf{H_m} \end{pmatrix}}_{\mathbf{T_a}} x,$$

The Laplacian Pyramid

Analysis operator

$$\underbrace{\begin{pmatrix} y_0 \\ y_1 \end{pmatrix}}_{y} = \underbrace{\begin{pmatrix} \mathbf{H_m} \\ \mathbf{I} - \mathbf{G}\mathbf{H_m} \end{pmatrix}}_{\mathbf{T_a}} x,$$

Do, Vetterli, Framing Pyramids, IEEE TSP, 2003

The Laplacian Pyramid

Analysis operator

$$\underbrace{\begin{pmatrix} y_0 \\ y_1 \end{pmatrix}}_{y} = \underbrace{\begin{pmatrix} \mathbf{H_m} \\ \mathbf{I} - \mathbf{G}\mathbf{H_m} \end{pmatrix}}_{\mathbf{T_a}} x,$$

Simple (traditional) left inverse

$$\hat{x} = \underbrace{\begin{pmatrix} \mathbf{G} & \mathbf{I} \end{pmatrix}}_{\mathbf{T_s}} \underbrace{\begin{pmatrix} y_0 \\ y_1 \end{pmatrix}}_{y}$$

$\mathbf{T_s}\mathbf{T_a} = \mathbf{I}$ with no conditions on $\mathbf{H}$ or $\mathbf{G}$

Do, Vetterli, Framing Pyramids, IEEE TSP, 2003

The Laplacian Pyramid

Pseudo Inverse ?

$$\mathbf{T_a}^{\dagger} = \left(\mathbf{T_a}^T \mathbf{T_a}\right)^{-1} \mathbf{T_a}^T$$

Let's try to use only filters

The Laplacian Pyramid

Pseudo Inverse ?

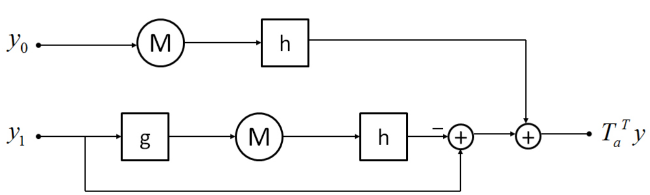
$$\mathbf{T_a}^{\dagger} = \left(\mathbf{T_a}^T \mathbf{T_a}\right)^{-1} \mathbf{T_a}^T$$

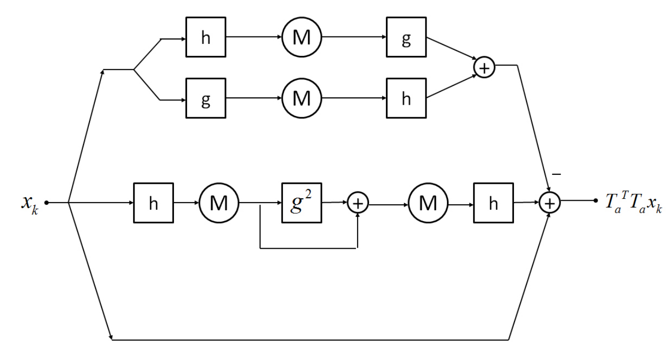
Let's try to use only filters

Landweber iterations involve only filters:

$$\arg\min_x \|\mathbf{T_a} x - y\|_2^2 \longrightarrow \hat{x}_{k+1} = \hat{x}_k + \tau \mathbf{T_a}^T (y - \mathbf{T_a} \hat{x}_k)$$

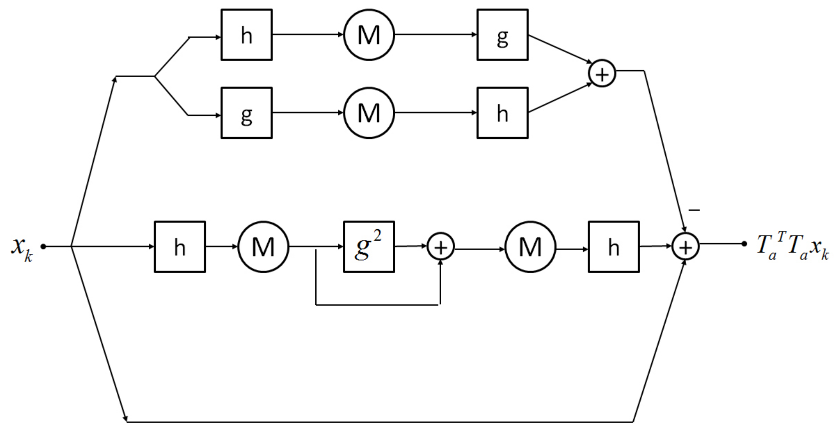
$$\mathbf{T_a}^T = (\mathbf{H_m}^T \quad \mathbf{I} - \mathbf{H_m}^T \mathbf{G}^T)$$





The Laplacian Pyramid

we can easily implement $\mathbf{T_a}^T\mathbf{T_a}$ with filters and masks:



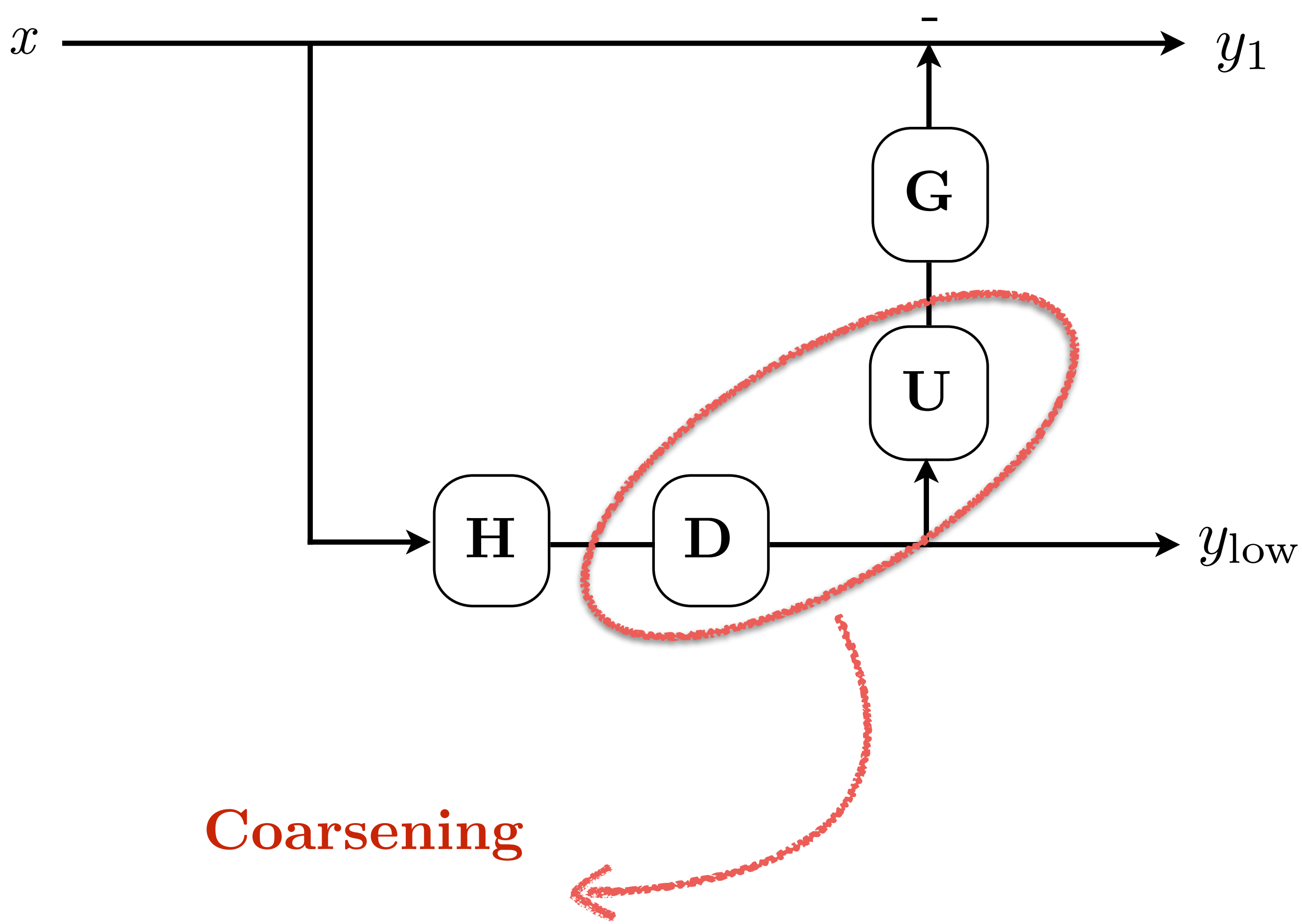
With the real symmetric matrix $\mathbf{Q} = \mathbf{T_a}^T\mathbf{T_a}$ and $b = \mathbf{T_a}^T y$

$$x_N = \tau \sum_{j=0}^{N-1} (\mathbf{I} - \tau\mathbf{Q})^j b$$

Use Chebyshev approximation of: $L(\omega) = \tau \sum_{j=0}^{N-1} (1 - \tau\omega)^j$

The Laplacian Pyramid

Analysis operator



Kron Reduction

In order to iterate the construction, we need to construct a graph on the reduced vertex set.

$$\mathbf{A}_{\mathrm{r}} = \mathbf{A}[\alpha,\alpha] - \mathbf{A}[\alpha,\alpha)\mathbf{A}(\alpha,\alpha)^{-1}\mathbf{A}(\alpha,\alpha]$$

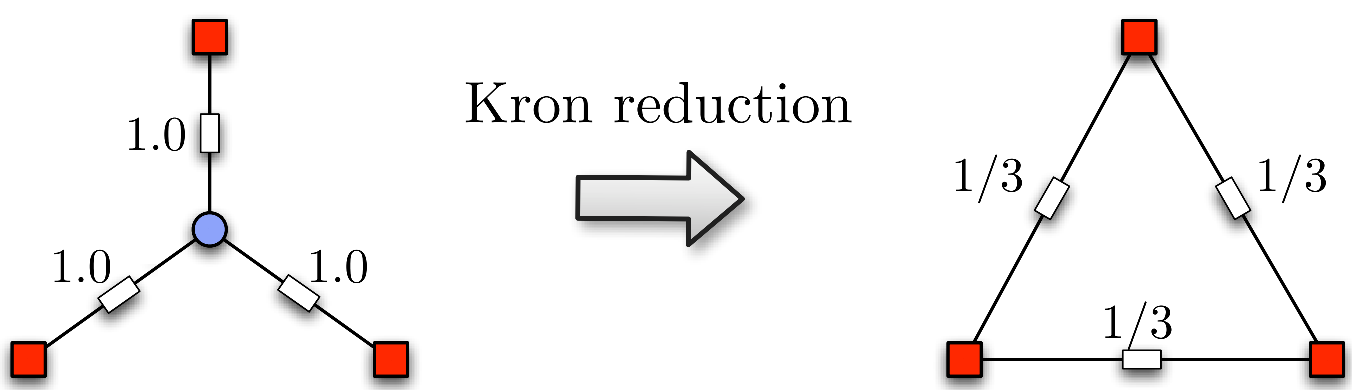
$$\mathbf{A} = \begin{bmatrix} \mathbf{A}[\alpha,\alpha] & \mathbf{A}[\alpha,\alpha) \\ \mathbf{A}(\alpha,\alpha] & \mathbf{A}(\alpha,\alpha) \end{bmatrix}$$

Kron Reduction

In order to iterate the construction, we need to construct a graph on the reduced vertex set.

$$\mathbf{A}_\mathrm{r} = \mathbf{A}[\alpha,\alpha] - \mathbf{A}[\alpha,\alpha)\mathbf{A}(\alpha,\alpha)^{-1}\mathbf{A}(\alpha,\alpha]$$

$$\mathbf{A} = \left[\begin{array}{cc} \mathbf{A}[\alpha,\alpha] & \mathbf{A}[\alpha,\alpha) \\ \mathbf{A}(\alpha,\alpha] & \mathbf{A}(\alpha,\alpha) \end{array}\right]$$



[Dorfler et al, 2011]

Kron Reduction

In order to iterate the construction, we need to construct a graph on the reduced vertex set.

$$\mathbf{A}_{\mathrm{r}} = \mathbf{A}[\alpha,\alpha] - \mathbf{A}[\alpha,\alpha)\mathbf{A}(\alpha,\alpha)^{-1}\mathbf{A}(\alpha,\alpha]$$

$$\mathbf{A} = \left[\begin{array}{cc} \mathbf{A}[\alpha,\alpha] & \mathbf{A}[\alpha,\alpha) \\ \mathbf{A}(\alpha,\alpha] & \mathbf{A}(\alpha,\alpha) \end{array}\right]$$

Properties: maps a weighted undirected laplacian to a weighted undirected laplacian

spectral interlacing (spectrum does not degenerate)

$$\lambda_k(\mathbf{A}) \leq \lambda_k(\mathbf{A}_r) \leq \lambda_{k+n-|\alpha|}(\mathbf{A})$$

disconnected vertices linked in reduced graph IFF there is a path that runs only through eliminated nodes

Example

Note: For a k-regular bipartite graph

$$\mathbf{L} = \begin{bmatrix} k\mathbf{I}_n & -\mathbf{A} \\ -\mathbf{A}^T & k\mathbf{I}_n \end{bmatrix}$$

Kron-reduced Laplacian: $\mathbf{L}_r = k^2\mathbf{I}_n - \mathbf{A}\mathbf{A}^T$

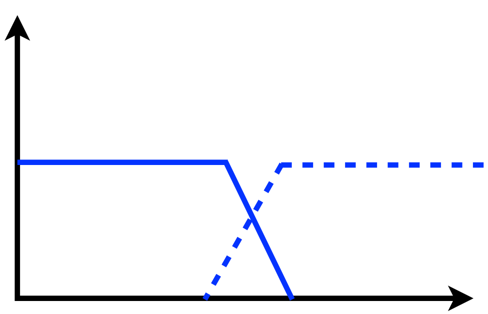
Example

Note: For a k-regular bipartite graph

$$\mathbf{L} = \begin{bmatrix} k\mathbf{I}_n & -\mathbf{A} \\ -\mathbf{A}^T & k\mathbf{I}_n \end{bmatrix}$$

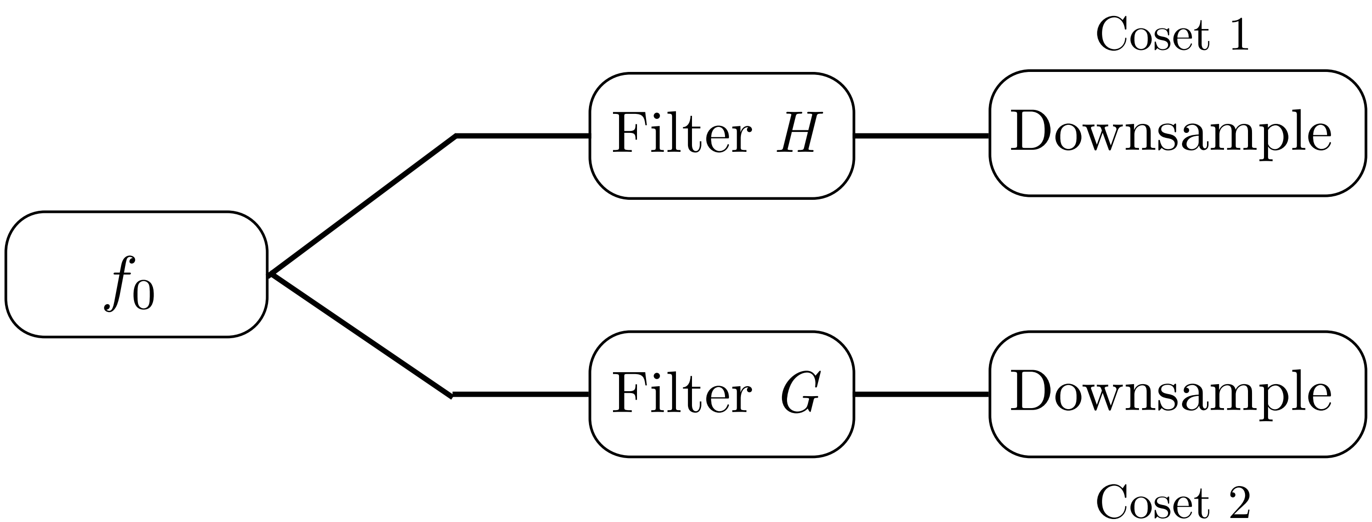
Kron-reduced Laplacian: $\mathbf{L}_r = k^2\mathbf{I}_n - \mathbf{A}\mathbf{A}^T$

$$\hat{f}_r(i) = \hat{f}(i) + \hat{f}(N-i) \quad i = 1, ..., N/2$$



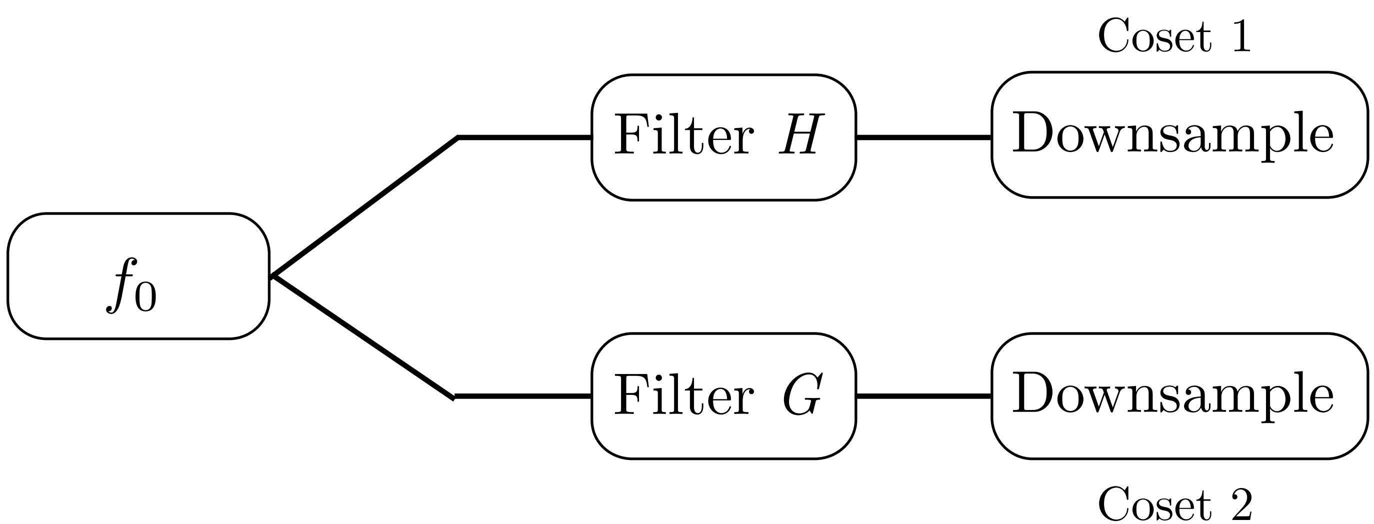
Filter Banks

2 critically sampled channels



Filter Banks

2 critically sampled channels

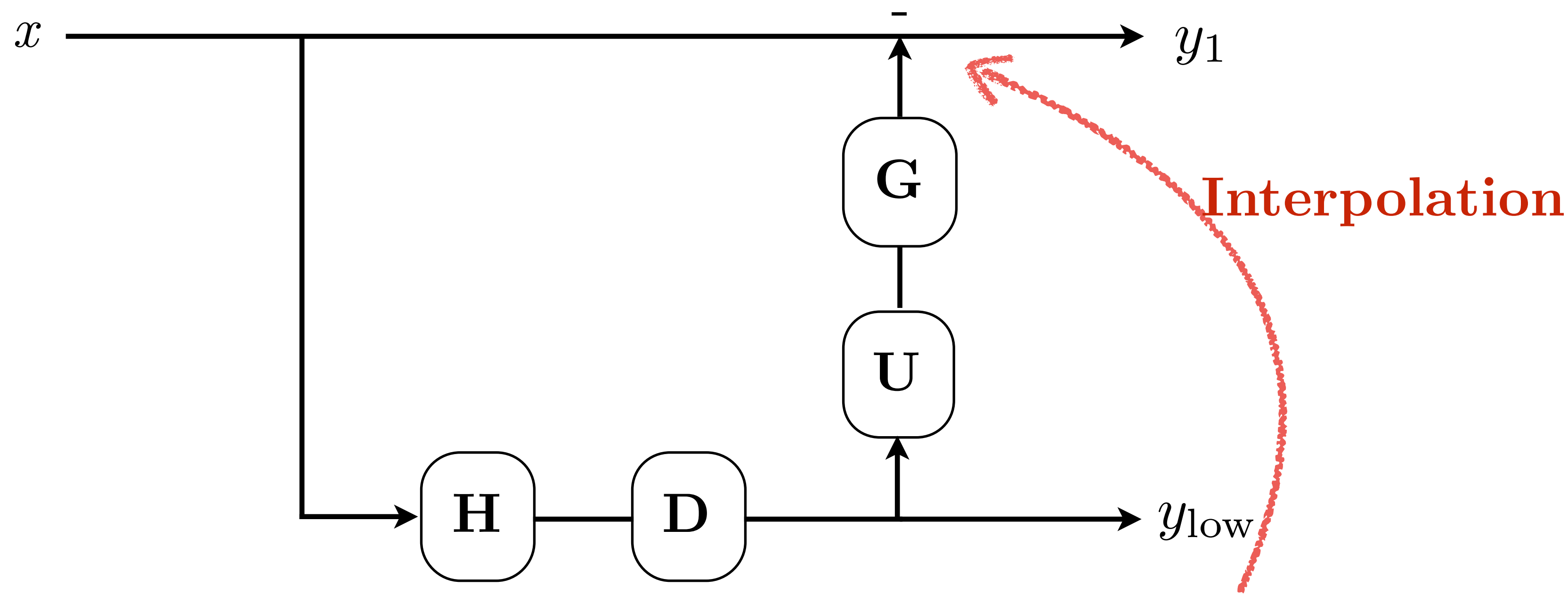


Theorem: For a k-RBG, the filter bank is perfect-reconstruction IFF

$$|H(i)|^2 + |G(i)|^2 = 2$$

$$H(i)G(N-i) + H(N-i)G(i) = 0$$

Reduction-aware interpolation



Idea: Optimize interpolation for reduction:

$$y[u] = \sum_{v \in V_1} \alpha[v] \varphi^v[u]$$ Shifted Green's function of L at vertex v

$$y[v'] = \sum_{v \in V_1} \alpha[v] \varphi^v[v'] = x[v'] \quad \forall v' \in V_1$$

Spline-like interpolation

Simple linear model:

$$f_{\text{interp}}(i) = \sum_{j \in \mathcal{V}_r} \alpha[j]\varphi_j(i) \qquad\qquad f_{\text{interp}} = \mathbf{\Phi}\alpha$$

With: $\varphi_j(i) = \big(T_j\varphi\big)(i) \qquad \mathbf{\Phi}[i,j] = \varphi_i(j)$

Spline-like interpolation

Simple linear model:

$$f_{\mathrm{interp}}(i) = \sum_{j \in \mathcal{V}_r} \alpha[j] \varphi_j(i) \qquad f_{\mathrm{interp}} = \mathbf{\Phi} \alpha$$

With: $\varphi_j(i) = \big(T_j \varphi\big)(i) \qquad \mathbf{\Phi}[i,j] = \varphi_i(j)$

Interpolation condition:

On the known vertices: $f_r = \mathbf{\Phi}_{\mathcal{V}_r} \alpha$

Solution depends on efficient, robust inversion of: $\alpha = \mathbf{\Phi}_{\mathcal{V}_r}^{-1} f_r$

Those weights can be computed using only filtering !

Spline-like interpolation

Regularized Laplacian: $$\tilde{\mathcal{L}} = \mu^{-1}\mathcal{L} + \mathbf{I}_{|\mathcal{V}|}$$

Stable pseudo-inverse: $$\tilde{\mathcal{L}}^{-1}[i,j] = \sum_{\ell=0}^{|\mathcal{V}|-1} \frac{1}{1+\mu^{-1}\lambda_\ell} u_\ell(i) u_\ell(j)$$

Spline-like interpolation

Regularized Laplacian: $\tilde{\mathcal{L}} = \mu^{-1}\mathcal{L} + \mathbf{I}_{|\mathcal{V}|}$

Stable pseudo-inverse: $\tilde{\mathcal{L}}^{-1}[i,j] = \sum_{\ell=0}^{|\mathcal{V}|-1} \frac{1}{1+\mu^{-1}\lambda_\ell} u_\ell(i) u_\ell(j)$

$$\varphi_j(i) = \sum_{\ell=0}^{|\mathcal{V}|-1} \frac{1}{1+\mu^{-1}\lambda_\ell} u_\ell(i) u_\ell(j)$$

Shifted Green's functions

Spline-like interpolation

Regularized Laplacian: $\tilde{\mathcal{L}} = \mu^{-1}\mathcal{L} + \mathbf{I}_{|\mathcal{V}|}$

Stable pseudo-inverse: $\tilde{\mathcal{L}}^{-1}[i,j] = \sum_{\ell=0}^{|\mathcal{V}|-1} \frac{1}{1+\mu^{-1}\lambda_\ell} u_\ell(i) u_\ell(j)$

$$\varphi_j(i) = \sum_{\ell=0}^{|\mathcal{V}|-1} \frac{1}{1+\mu^{-1}\lambda_\ell} u_\ell(i) u_\ell(j)$$

Shifted Green's functions

$$\begin{aligned} \tilde{\mathcal{L}}_r f_{\text{interp}}(i) &= \tilde{\mathcal{L}}_r f_r(i), \forall i \in \mathcal{V}_r \\ &= \sum_{j \in \mathcal{V}_r} \alpha[j] (\tilde{\mathcal{L}}_r \varphi_j)(i) \end{aligned}$$

Note:

$$\begin{aligned} \tilde{\mathcal{L}}\varphi_j(i) &= \tilde{\mathcal{L}}\tilde{\mathcal{L}}^{-1}\delta_j(i) \\ &= \delta_j(i) \end{aligned}$$

Does this property carry over to the Kron reduced Laplacian?

Spline-like interpolation

Lemma: Inversion/Reduction commute for the (regularized) Laplacian

$$\left(\tilde{\mathcal{L}}^{-1}\right)_{\mathcal{V}_r} = \left(\tilde{\mathcal{L}}_r\right)^{-1}$$

This implies invariance of the Green's functions via reduction and therefore

$$\alpha = \tilde{\mathcal{L}}_r f_r \qquad f_{\mathrm{interp}} = \mathbf{\Phi}\alpha$$

Spline-like interpolation

Lemma: Inversion/Reduction commute for the (regularized) Laplacian

$$\left(\tilde{\mathcal{L}}^{-1}\right)_{\mathcal{V}_r} = \left(\tilde{\mathcal{L}}_r\right)^{-1}$$

This implies invariance of the Green's functions via reduction and therefore

$$\alpha = \tilde{\mathcal{L}}_r f_r \qquad f_{\mathrm{interp}} = \mathbf{\Phi}\alpha$$

Algorithm: Reduce graph

Apply reduced Laplacian to vertex data

Replace old data with newly calculated coefficients

Filter with Green's kernel

Sparsification

Kron reduction produces denser and denser graphs

After each reduction we use Spielman's sparsification algorithm to obtain an equivallent but sparser graph

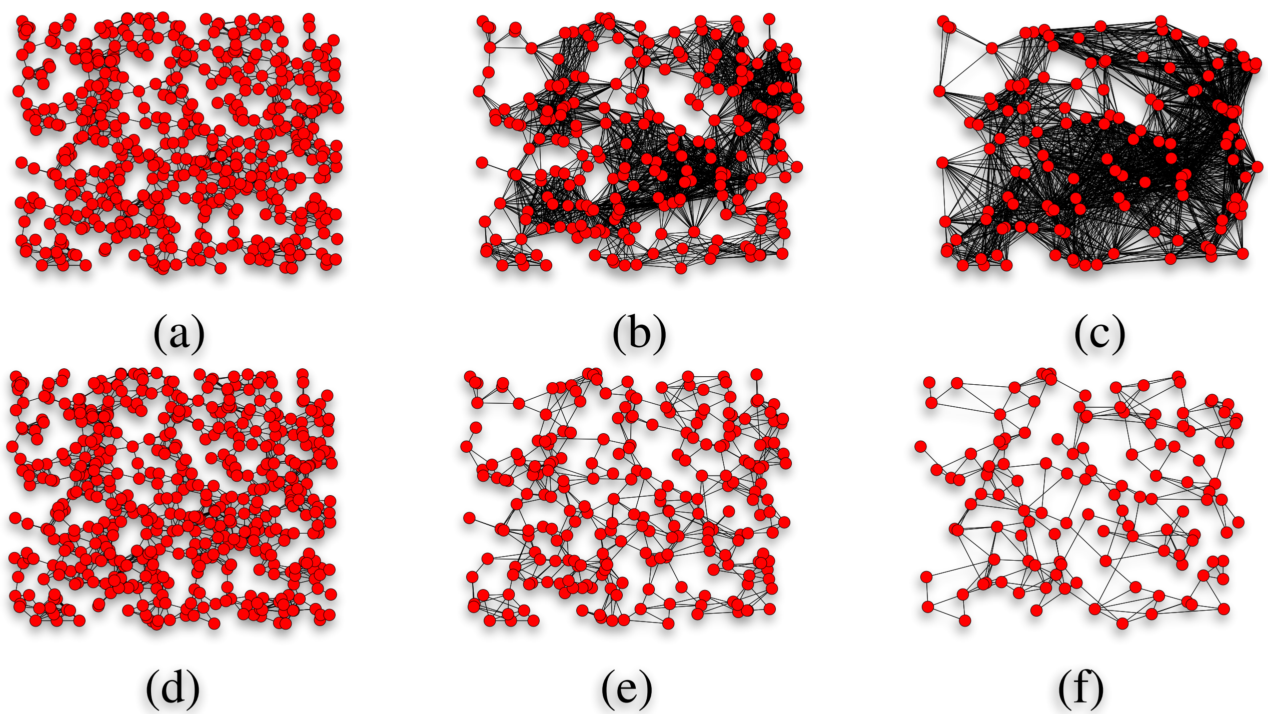
Approx preserves Laplacian quadratic form

Explicit control based on effective resistance of edges

Spielman and Srivastava, Graph sparsification by effective resistances, SIAM J. Comp, 2011

Sparsification

Kron reduction produces denser and denser graphs

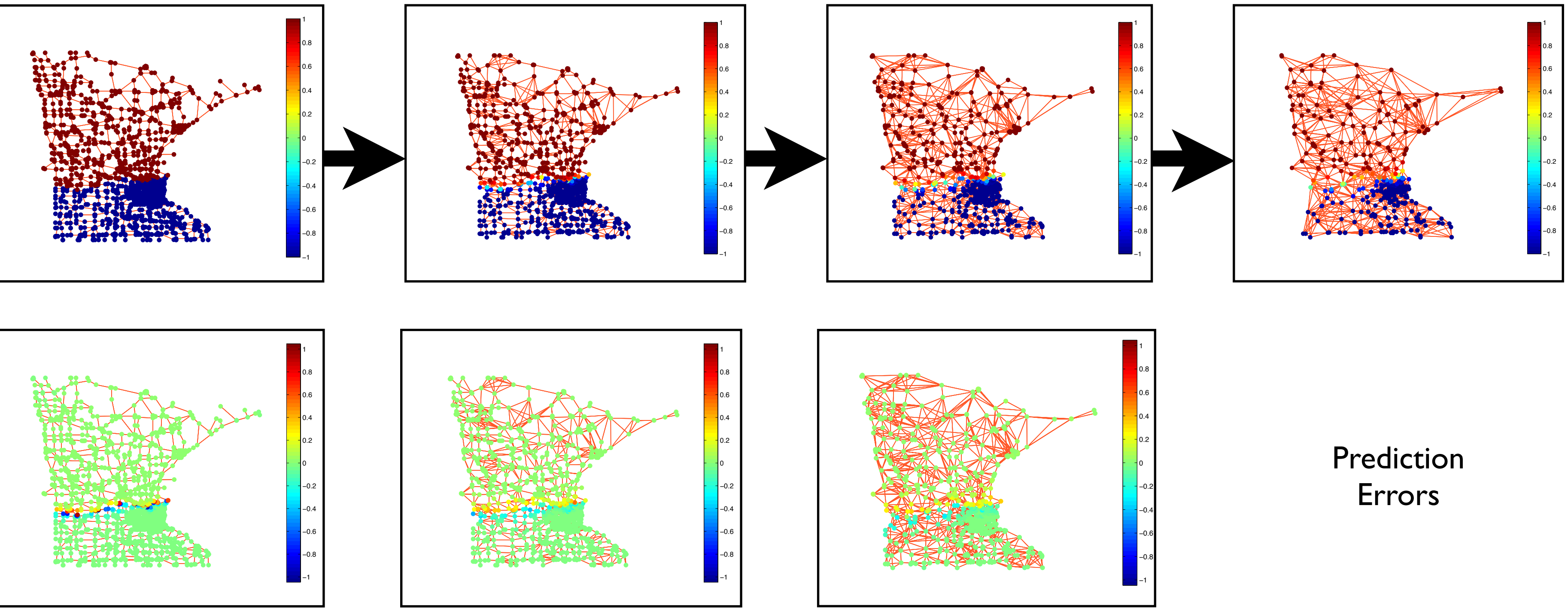


After ... rithm

Approx p... quadratic ... ed on effective resistance of edges

Spielman and Srivastava, Graph sparsification by effective resistances, SIAM J. Comp, 2011

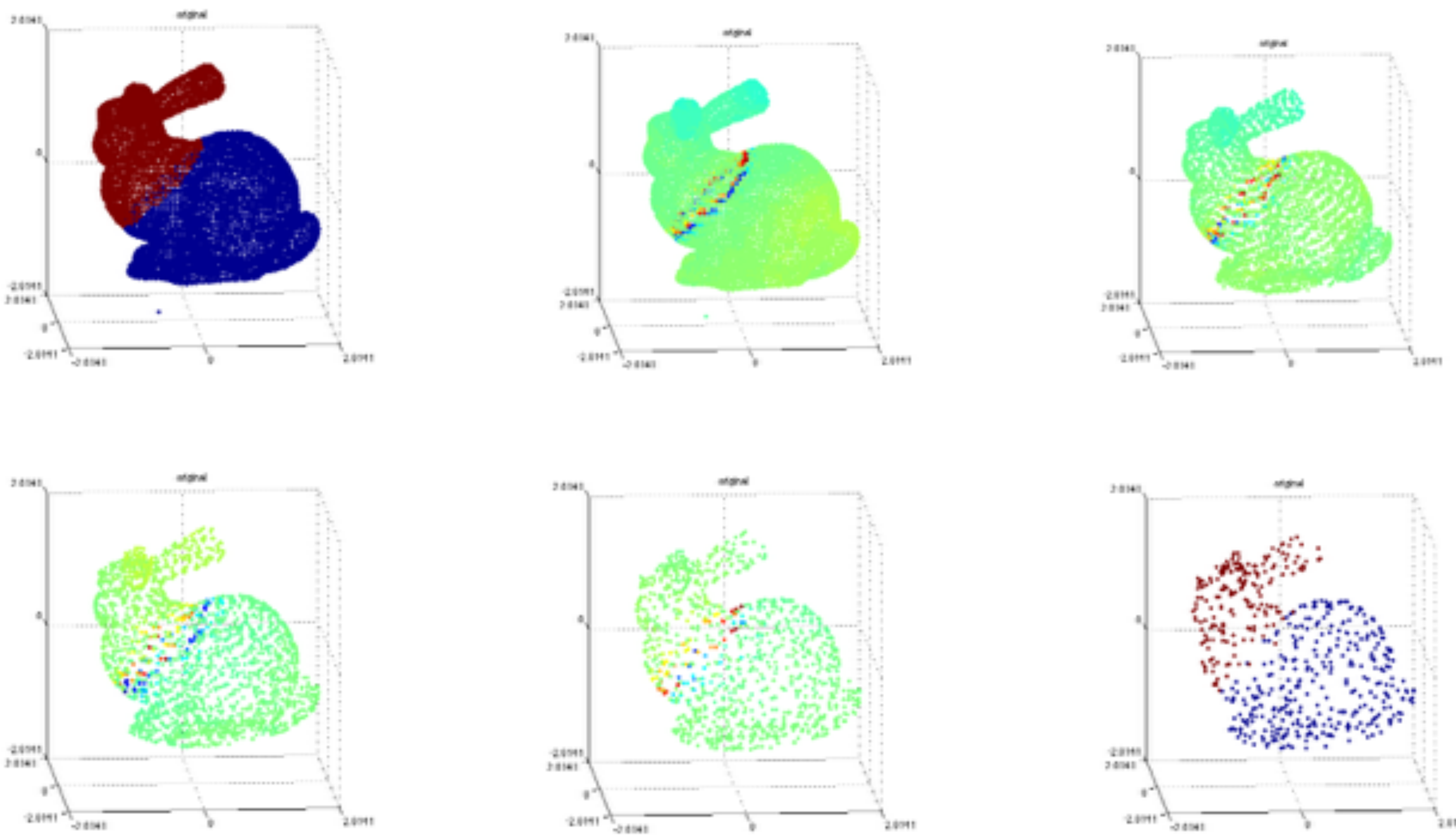
Example



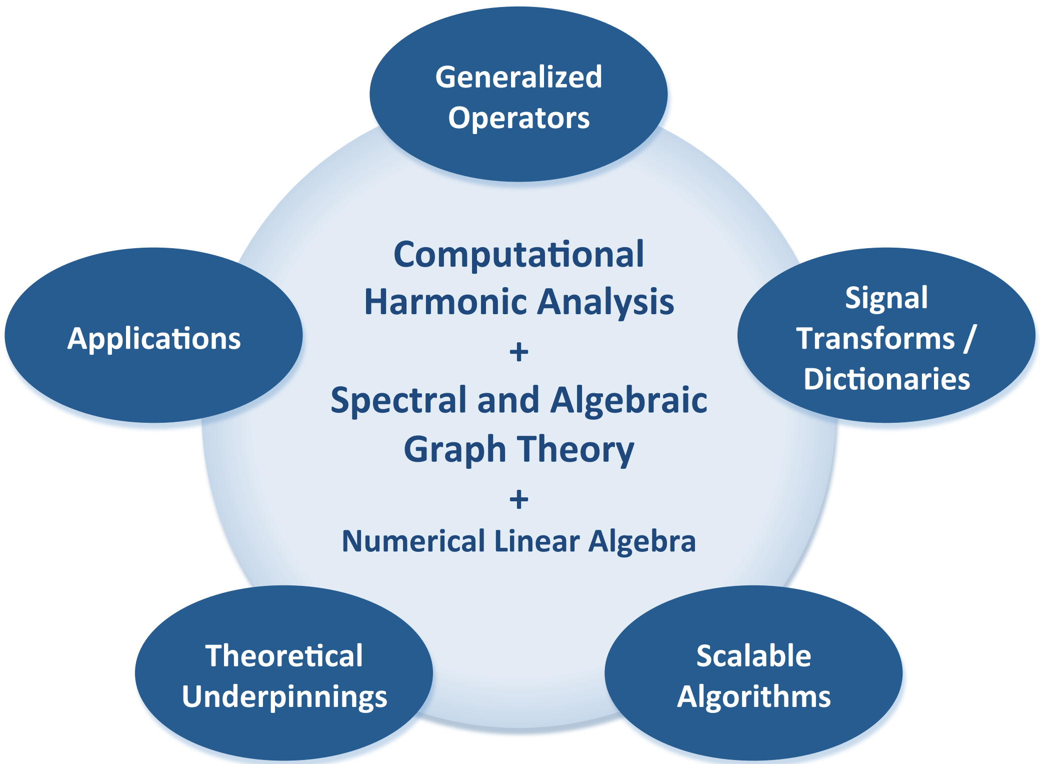
Coarse Approximations

Prediction Errors

2642 1334 669 337



Outlook



- Application of graph signal processing techniques to real science and engineering problems is in its infancy
- Theoretical connections between classes of graph signals, the underlying graph structure, and sparsity of transform coefficients