Ondelettes sur Graphes: Algorithmes et Applications

Pierre Vandergheynst

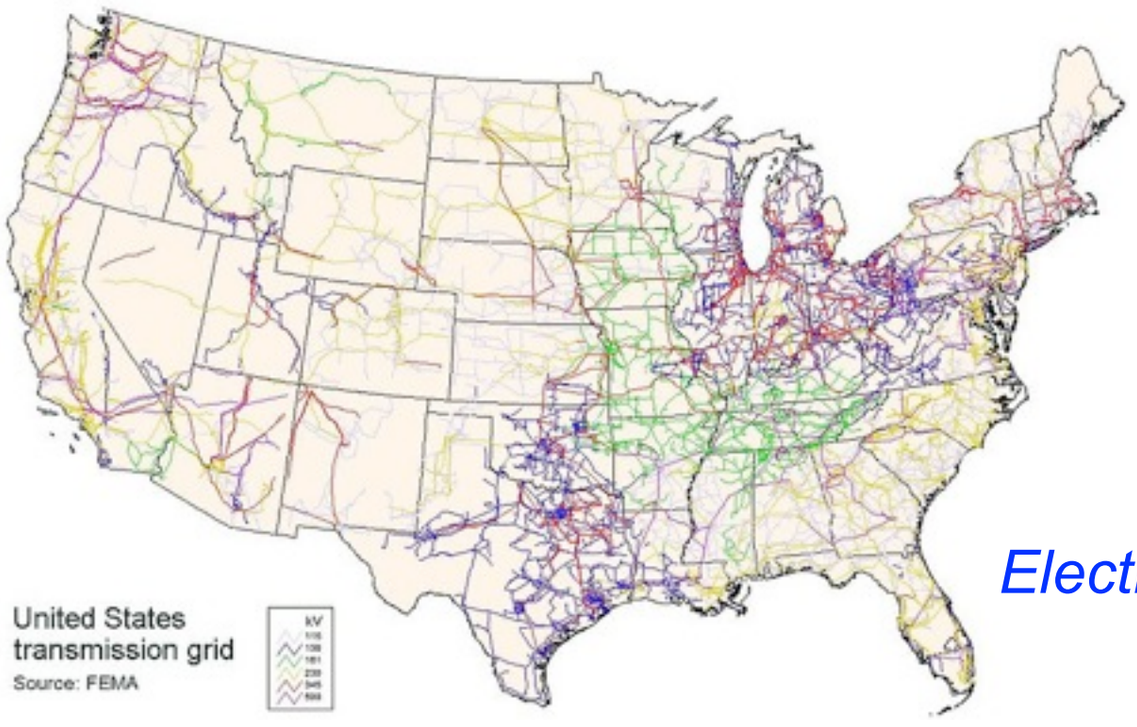
Signal Processing Lab, EPFL

Transformées Multirésolution Géométriques

GDR ISIS, 1er avril 2011

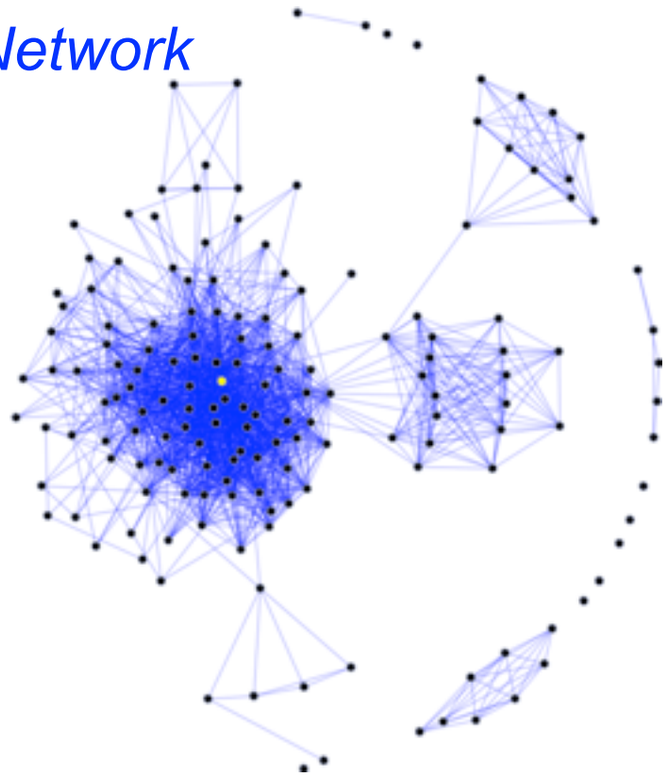
Ont collaboré: R. Gribonval, D. Hammond, D. Shuman

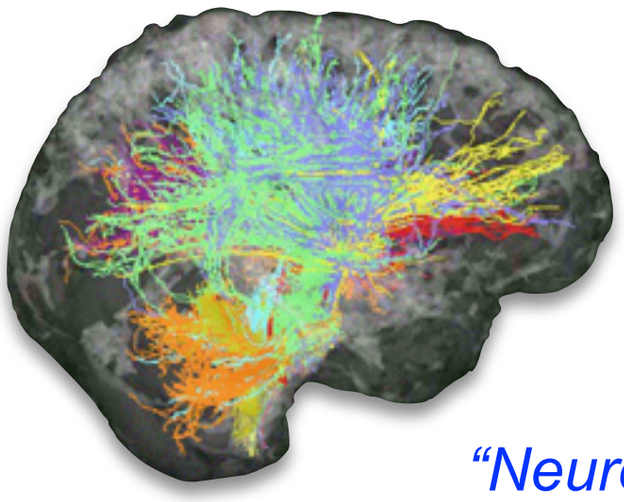
All sorts of networks around us



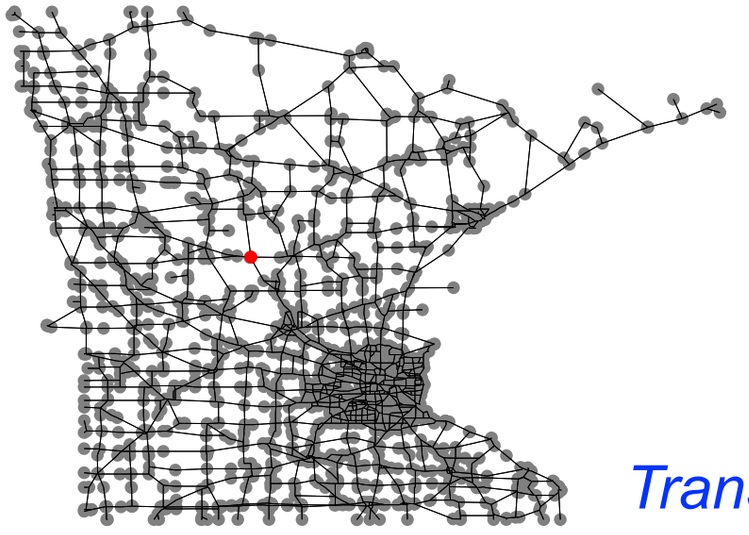
Electrical Network

Social Network





“Neuronal” Network



Transportation Network

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 $\quad |S| = l < n$

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

Questions:

How can unlabeled 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

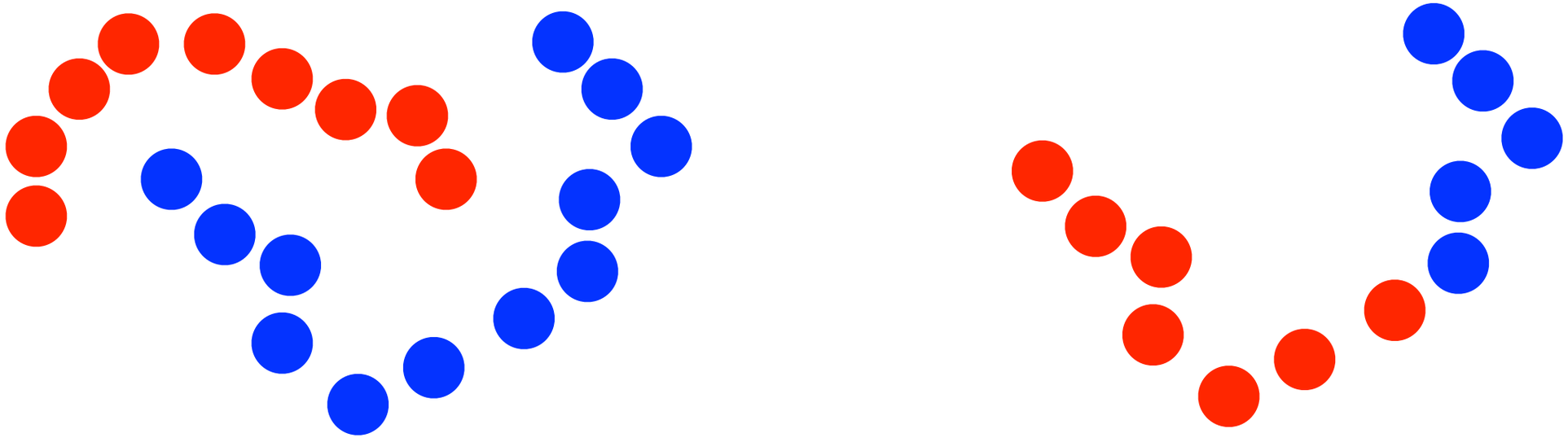
simplifies/stabilizes selected model

Learning on/with Graphs

How can unlabeled 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: $\Delta f = \sum_{i,j \in X} \mathbf{W}_{ij}(f(x_i) - f(x_j))^2$

$= \mathbf{f}^t L \mathbf{f}$ *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 ?

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

Wavelet Ingredients

Wavelet transform based on two operations:

Dilation (or scaling) and Translation (or localization)

$$\psi_{s,a}(x) = \frac{1}{s}\psi\left(\frac{x-a}{s}\right)$$

$$(T^s f)(a) = \int \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right) f(x)dx \qquad (T^s f)(a) = \langle \psi_{(s,a)}, f \rangle$$

Equivalently: $$(T^s \delta_a)(x) = \frac{1}{s}\psi^*\left(\frac{x-a}{s}\right)$$

$$(T^s f)(x) = \frac{1}{2\pi}\int e^{i\omega x}\hat{\psi}^*(s\omega)\hat{f}(\omega)d\omega$$

Graph Laplacian and Spectral Theory

$G = (V, E, w)$ weighted, undirected graph

Non-normalized Laplacian: $\mathcal{L} = D - A$ Real, symmetric

$$(\mathcal{L}f)(i) = \sum_{i \sim j} w_{i,j}(f(i) - f(j))$$

Why Laplacian ? $\mathbb{Z}^2$ with usual stencil

$$(\mathcal{L}f)_{i,j} = 4f_{i,j} - f_{i+1,j} - f_{i-1,j} - f_{i,j+1} - f_{i,j-1}$$

In general, graph laplacian from nicely sampled manifold converges to Laplace-Beltrami operator

Remark:

$$\mathcal{L}^{norm} = D^{-1/2}\mathcal{L}D^{-1/2} = I - D^{-1/2}AD^{-1/2}$$

Graph Laplacian and Spectral Theory

$$\frac{d^2}{dx^2} \Rightarrow e^{i\omega x} \Rightarrow f(x) = \frac{1}{2\pi}\int \hat{f}(\omega) e^{i\omega x} d\omega$$

Eigen decomposition of Laplacian: $\mathcal{L}\phi_l = \lambda_l \phi_l$

For simplicity assume connected graph and $0 = \lambda_0 < \lambda_1 \leq \lambda_2 ... \leq \lambda_{N-1}$

For any function on the vertex set (vector) we have:

$$\hat{f}(\ell) = \langle \phi_\ell, f \rangle = \sum_{i=1}^{N} \phi_\ell^*(i) f(i) \quad \text{Graph Fourier Transform}$$

$$f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell) \phi_\ell(i)$$

Spectral Graph Wavelets

Remember good old Euclidean case:

$$(T^s f)(x) = \frac{1}{2\pi} \int e^{i\omega x} \hat{\psi}^*(s\omega) \hat{f}(\omega) d\omega$$

We will adopt this operator view

Operator-valued function via continuous *Borel functional calculus*

$$g : \mathbb{R}^+ \to \mathbb{R}^+ \qquad T_g = g(\mathcal{L}) \quad \textbf{Operator-valued function}$$

Action of operator is induced by its Fourier symbol

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

Spectral Graph Wavelets

Now on to our main ingredients !

Dilation operates through operator: $T_g^t = g(t\mathcal{L})$

Translation (localization):

Define $\psi_{t,j} = T_g^t \delta_j$ response to a delta at vertex j

$$\psi_{t,j}(i) = \sum_{\ell=0}^{N-1} g(t\lambda_\ell)\phi_\ell^*(j)\phi_\ell(i)$$

$$\psi_{t,a}(u) = \int_{\mathbb{R}} d\omega\, \hat{\psi}(t\omega) e^{-j\omega a} e^{j\omega u}$$

And so formally define the graph wavelet coefficients of f:

$$W_f(t,j) = \langle \psi_{t,j}, f \rangle \qquad W_f(t,j) = T_g^t f(j) = \sum_{\ell=0}^{N-1} g(t\lambda_\ell)\hat{f}(\ell)\phi_\ell(j)$$

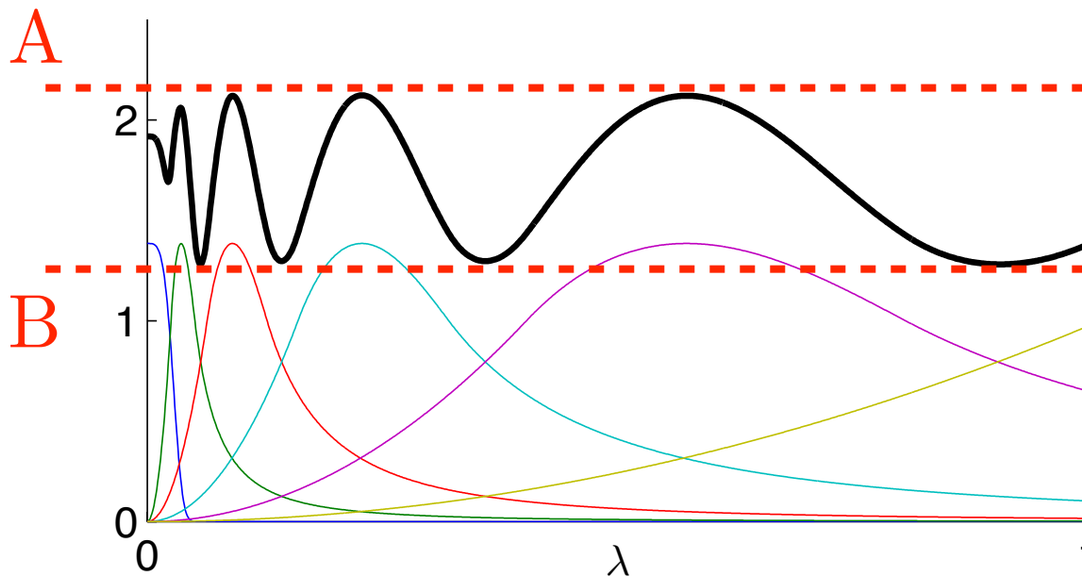
Frames

$$\exists\, A, B > O,\ \exists h : \mathbb{R}_+ \to \mathbb{R}_+ \text{ (i.e. scaling function)}$$

$$0 < A \leqslant h^2(u) + \sum_s g(t_s u)^2 \leqslant B < \infty$$

scaling function $h^2(u)$ — *wavelets* $g(t_s u)^2$

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



A simple way to get a tight frame:

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

for any admissible kernel g

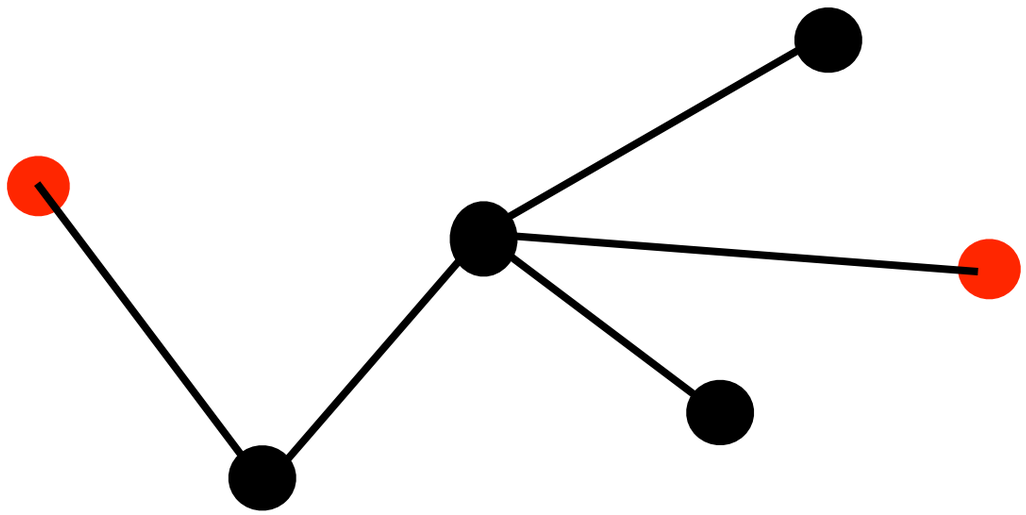
Scaling & Localization

Effect of operator dilation ?

Acts in subtle way, depends on kernel

but has universal “localization” effect

refaire mieux

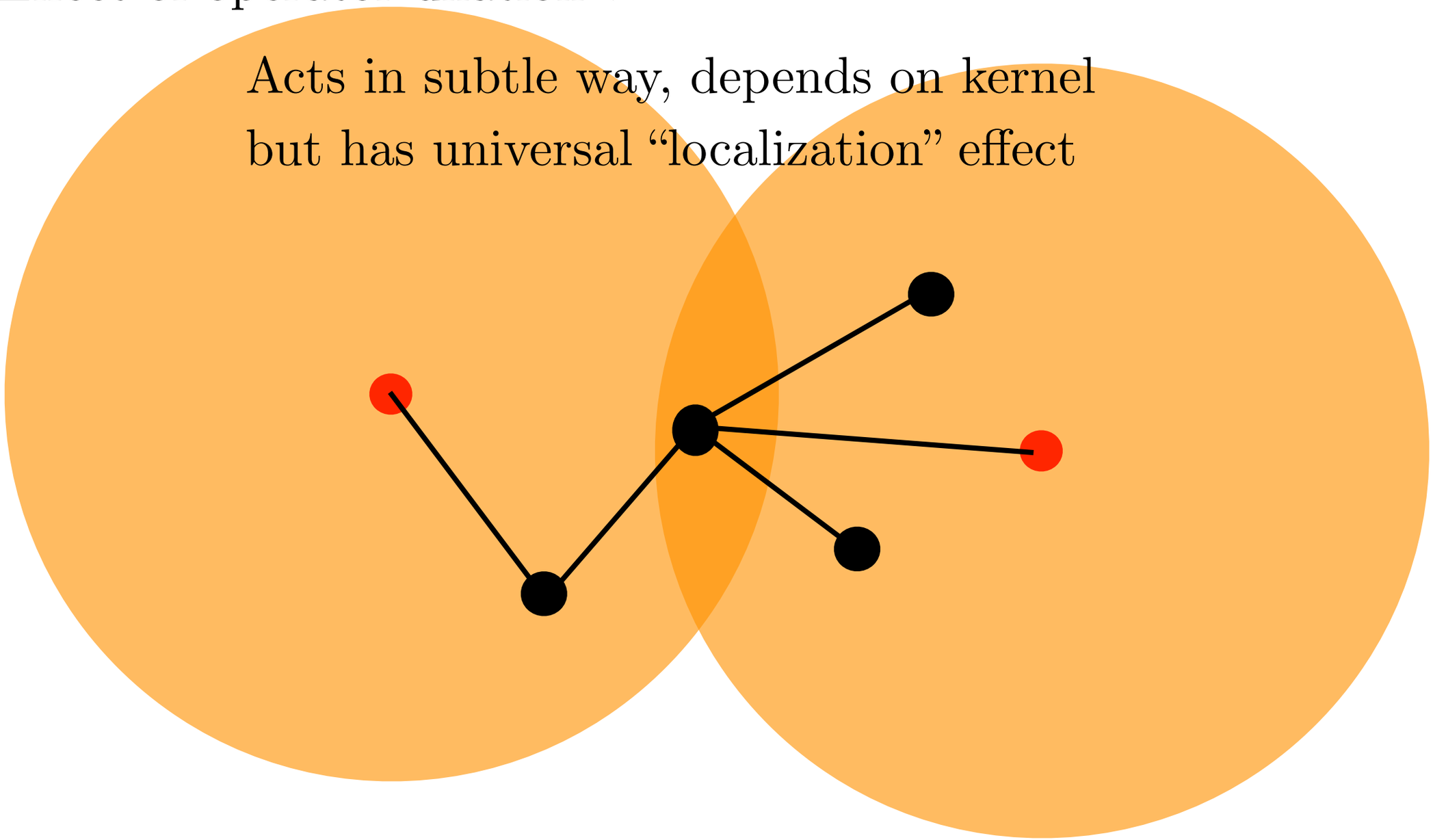


Scaling & Localization

Effect of operator dilation ?

Acts in subtle way, depends on kernel

but has universal “localization” effect

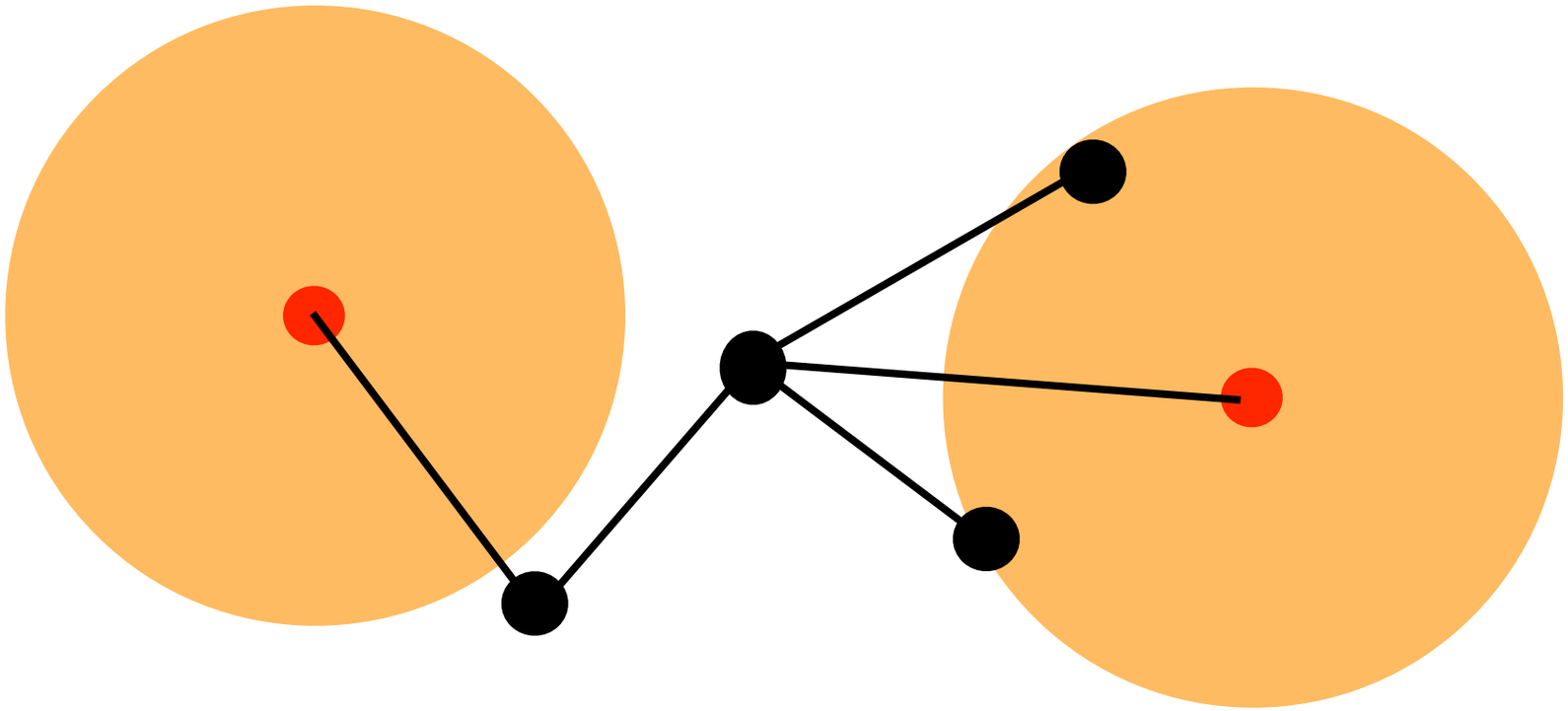


Scaling & Localization

Effect of operator dilation ?

Acts in subtle way, depends on kernel

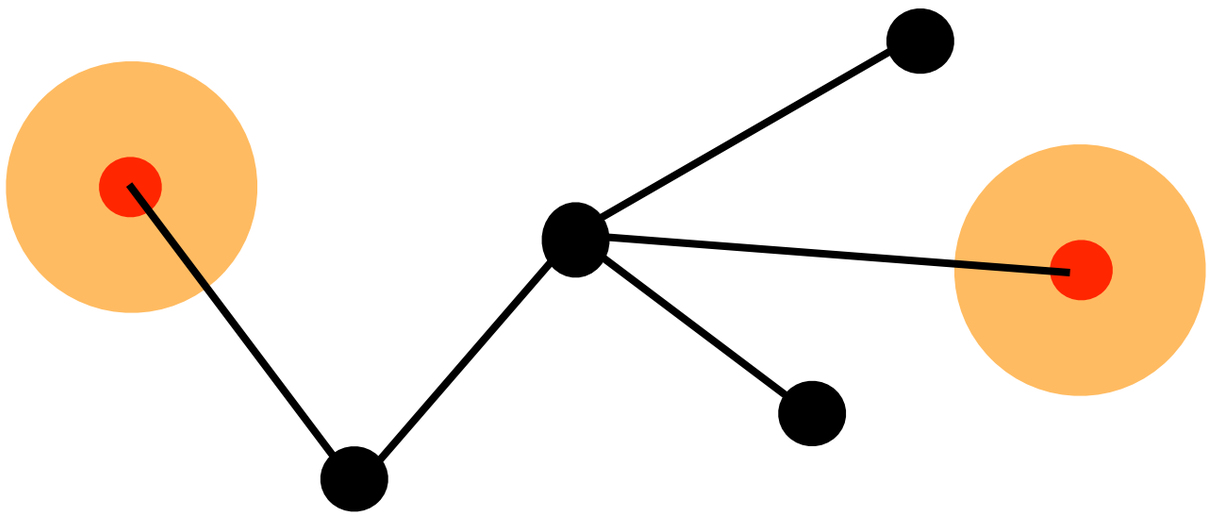
but has universal “localization” effect



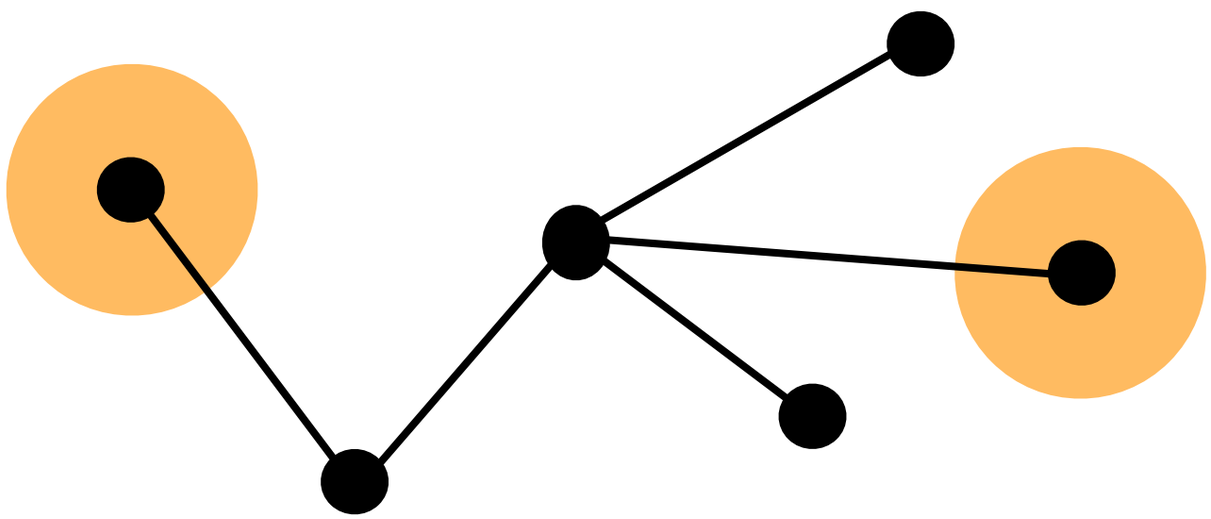
Scaling & Localization

Effect of operator dilation ?

Acts in subtle way, depends on kernel
but has universal “localization” effect



Scaling & Localization



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

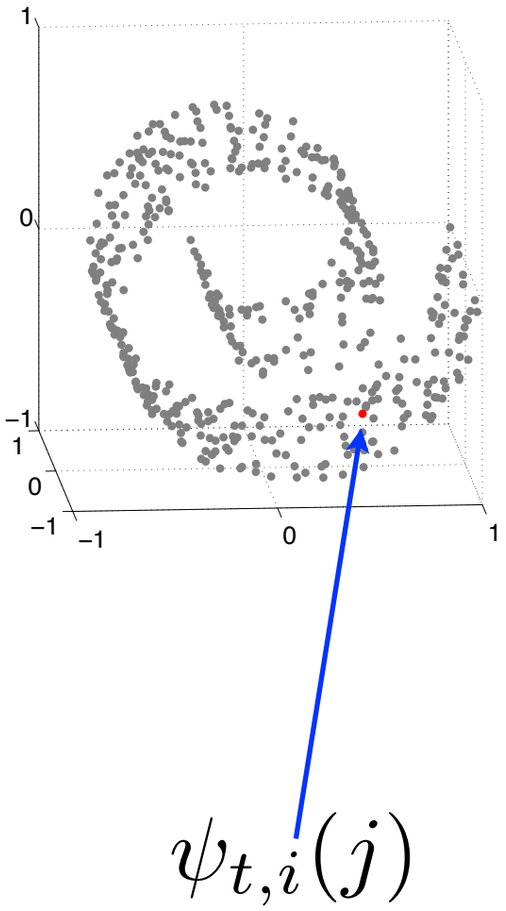
Study matrix element: $\psi_{t,i}(j) = \langle \psi_{t,i}, \delta_j \rangle = \langle T_g^t \delta_i, \delta_j \rangle$

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

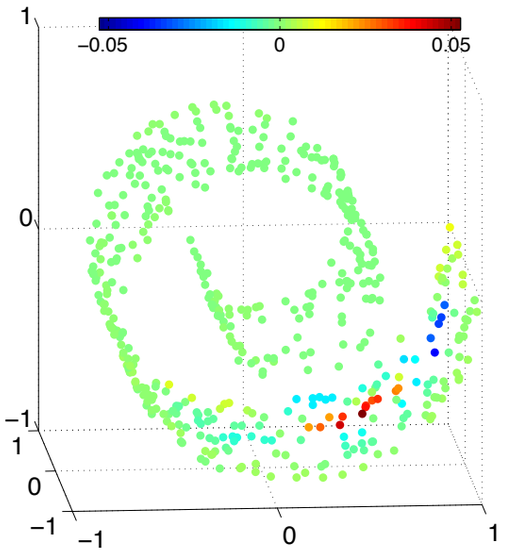
$$\frac{\psi_{t,j}(i)}{\|\psi_{t,j}\|} \leq Dt \quad \text{for any t smaller than a critical scale function of } d_G(i,j)$$

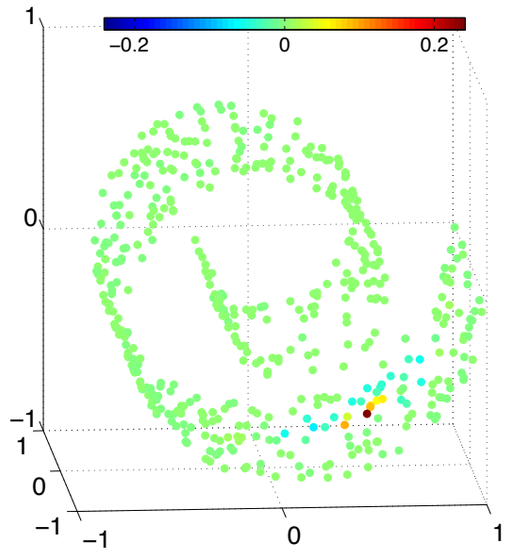
Reason ? At small scale, wavelet operator behaves like power of Laplacian

Scaling & Localization

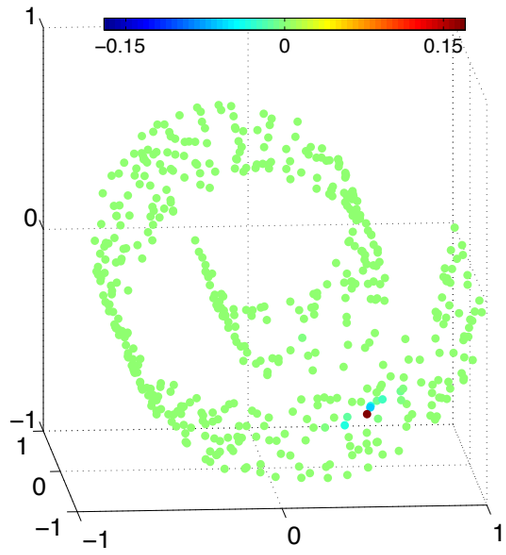


$\psi_{t,i}(j)$

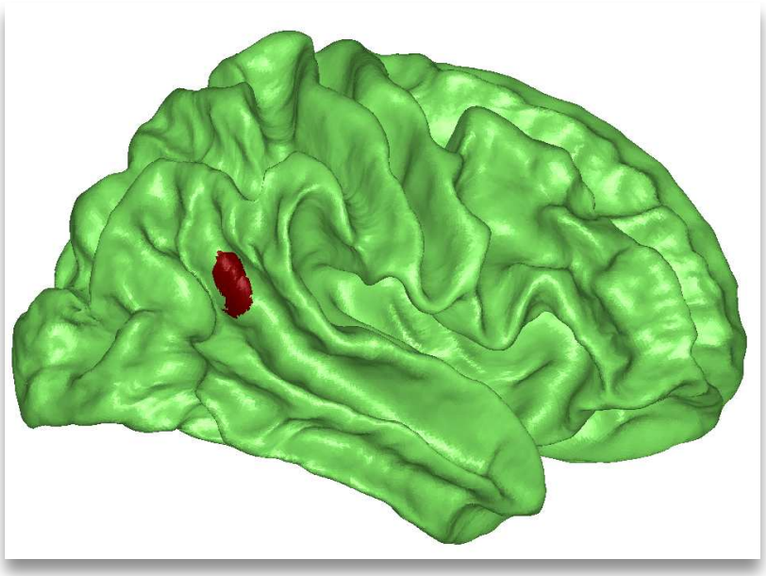


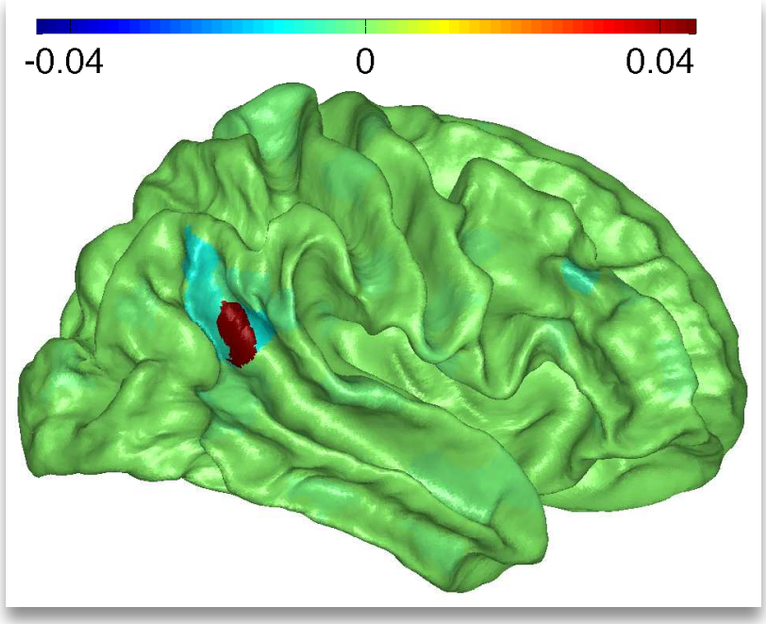


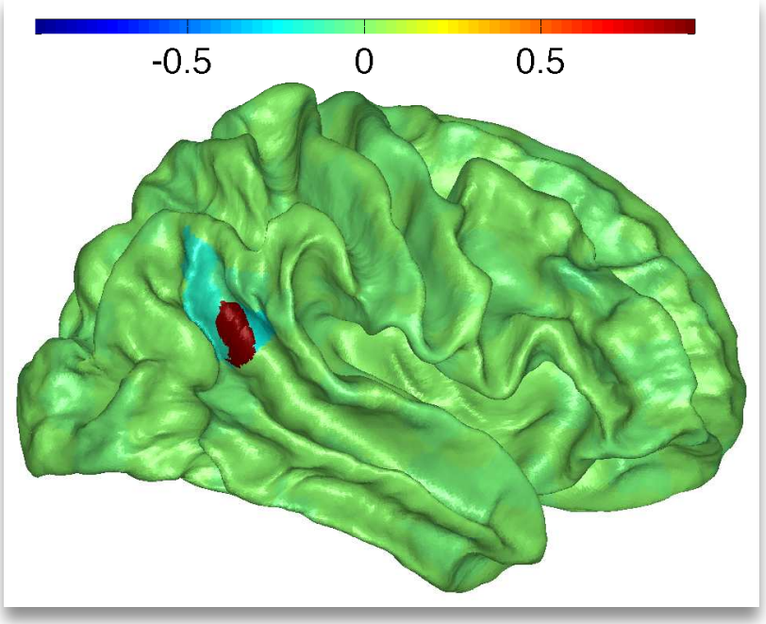
decreasing scale

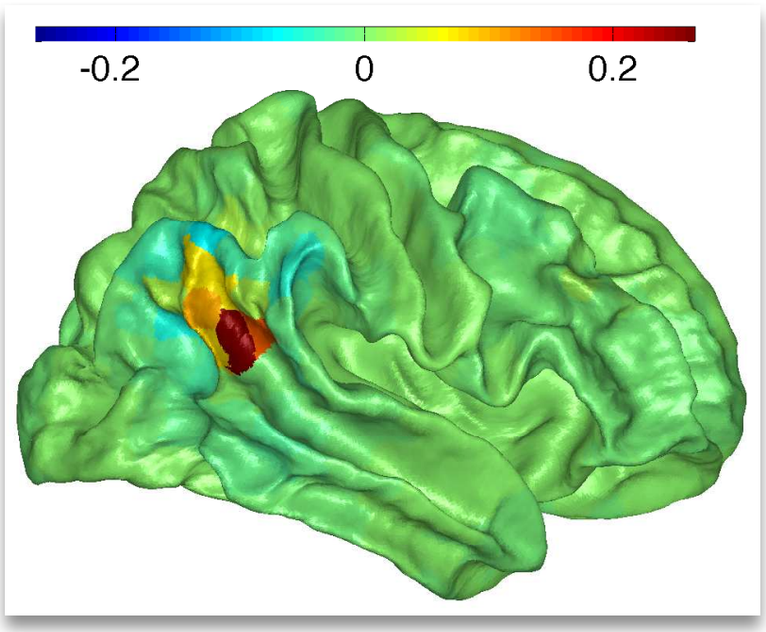


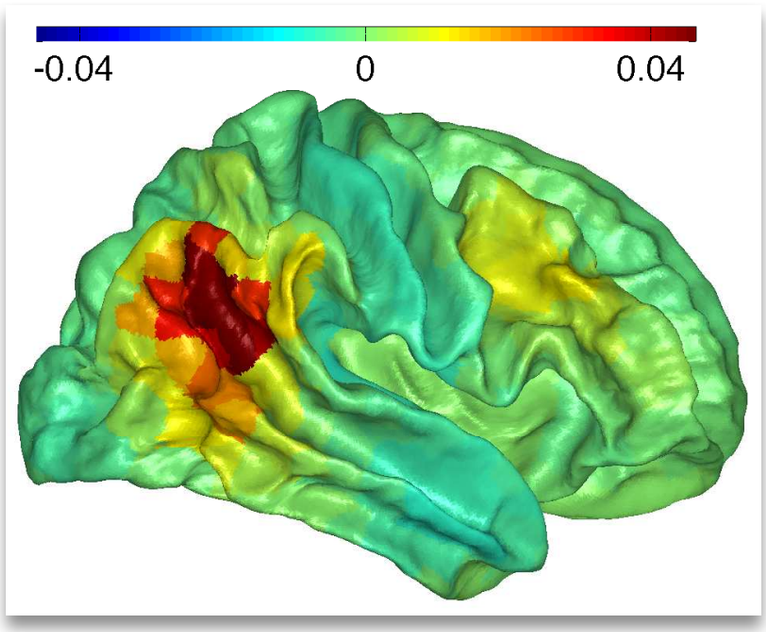
Example

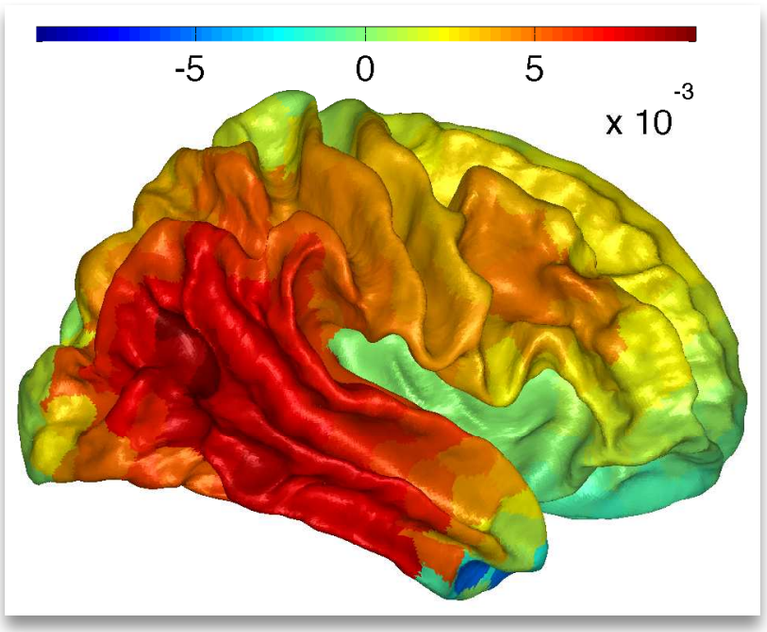












Sparsity and Smoothness on Graphs

Using a dictionary of graph wavelets, sparsity and smoothness on graph are the same thing !

Idea: for a "Meyer kernel" on the spectrum of G

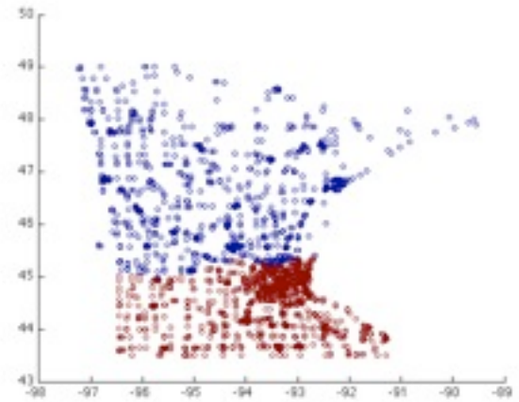
$$\sum_{i \in V} |\langle \psi_{2^{-j},i}, f \rangle|^2 = \sum_{l} |g(2^j \lambda_l)|^2 |\hat{f}(\lambda_l)|^2$$

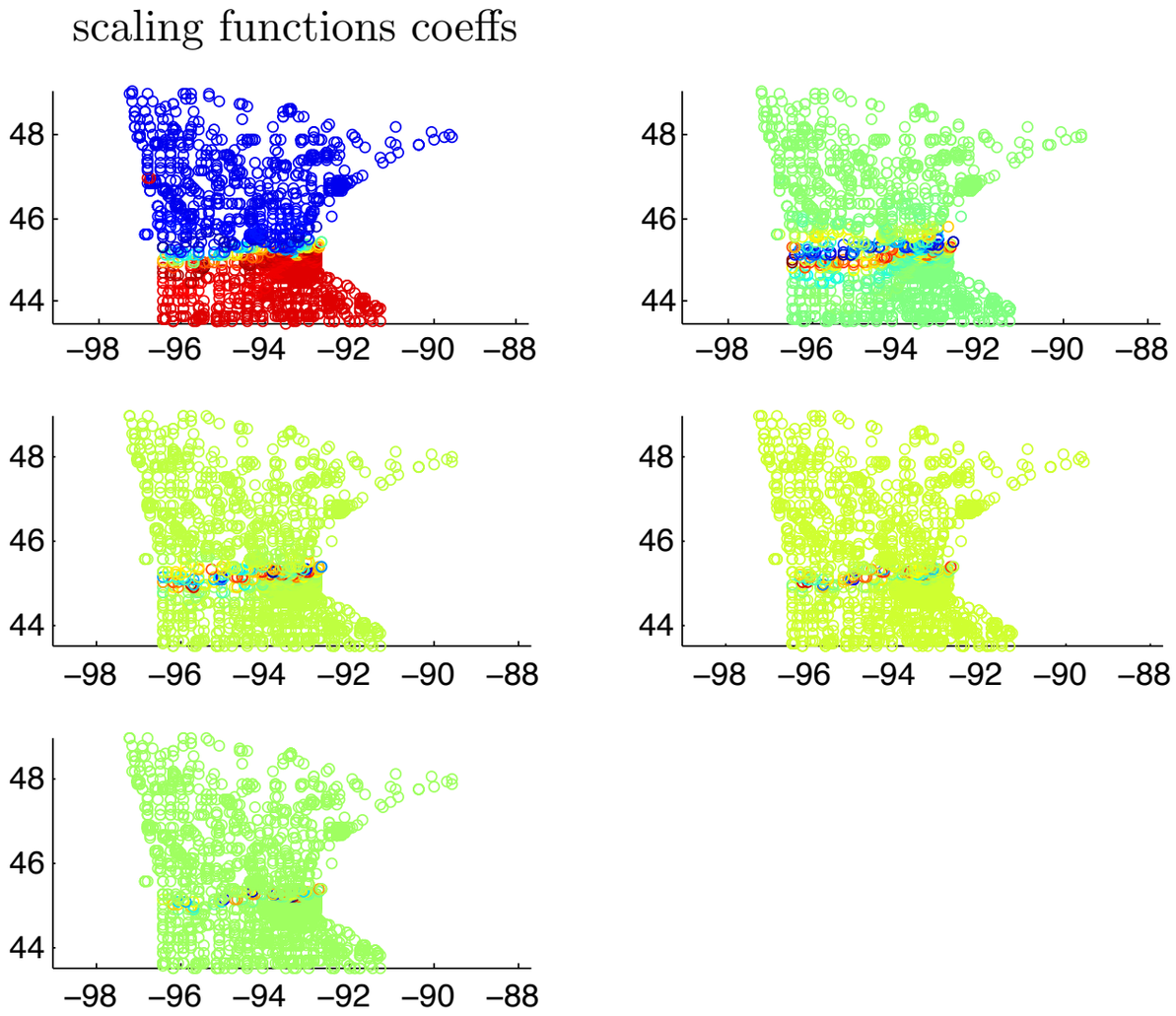
$$= \sum_{2^{-j-1}\lambda_{\max} \leq \lambda_l \leq 2^{-j}\lambda_{\max}} |\hat{f}(\lambda_l)|^2$$

$$A \sum_l \lambda_l^{2s} |\hat{f}(\lambda_l)|^2 \leq \sum_j 2^{-2sj} \sum_i |\langle \psi_{2^{-j},i}, f \rangle|^2 \leq B \sum_l \lambda_l^{2s} |\hat{f}(\lambda_l)|^2$$

$\|f\|^2_{G,2s} = \sum_l \lambda_l^{2s} |\hat{f}(\lambda_l)|^2$ discrete Sobolev semi-norm on G

Sparsity and Smoothness on Graphs



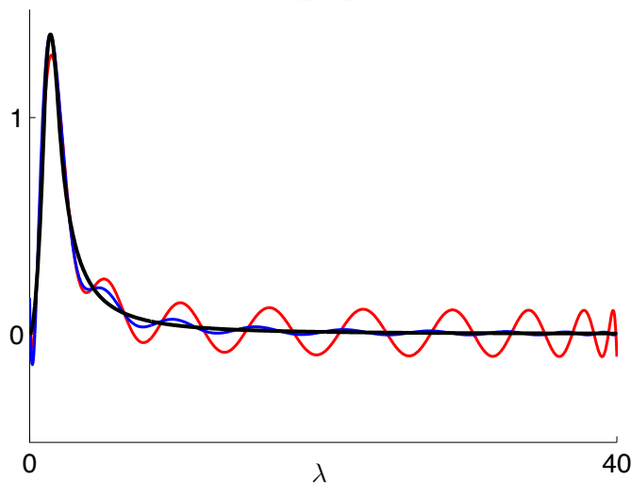


Remark on Implementation

Not necessary to compute spectral decomposition

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

ex: Chebyshev, minimax



Then wavelet operator expressed with powers of Laplacian:

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

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 Chebyshef)

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

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

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

In particular $O(\sum_{n=1} M_n|E|)$

Note: "same" algorithm for adjoint !

Distributed Computation

Scenario: Network of N nodes, each knows

- local data f(n)
- local neighbors
- M Chebyshev coefficients of wavelet kernel
- A global upper bound on largest eigenvalue of graph laplacian

To compute: $\left(\tilde{\mathbf{\Phi}} f\right)_{(j-1)N+n} = \left(\frac{1}{2} c_{j,0} f + \sum_{k=1}^{M} c_{j,k} \overline{T}_k(\mathcal{L}) f\right)_n$

$\left(\overline{T}_1(\mathcal{L}) f\right)_n = \left(\frac{2}{\alpha}(\mathcal{L} - \alpha I) f\right)_n$ **sensor only needs f(n) from its neighbors**

$\left(\overline{T}_k(\mathcal{L}) f\right) = \frac{2}{\alpha}(\mathcal{L} - \alpha I)\left(\overline{T}_{k-1}(\mathcal{L}) f\right) - \overline{T}_{k-2}(\mathcal{L}) f$ **Computed by exchanging last computed values**

Distributed Computation

Communication cost: 2M|E| messages of length 1 per node

Example: distributed denoising, or distributed regression, with Lasso

$$\arg\min_a \frac{1}{2}\|y - \mathbf{\Phi}^* a\|_2^2 + \|a\|_{1,\mu}$$

$$a_i^{(k)} = \mathcal{S}_{\mu_i,\tau}\left(\left[a^{k-1} + \tau\mathbf{\Phi}(y - \mathbf{\Phi}^* a^{k-1})\right]_i\right)$$

$$\mathcal{S}_{\mu_i\tau}(z) := \begin{cases} 0 & , \text{ if } \mid z \mid \leq \mu_i\tau \\ z - \text{sgn}(z)\mu_i\tau & , \text{ o.w.} \end{cases}$$

Total communication cost:

Distributed Lasso [Mateos, Bazerque, Gianakis] Cost $\sim |E|N$

Chebyshev $\mathbf{\Phi}y$ 2M|E| messages of length 1

$\mathbf{\Phi}\mathbf{\Phi}^* a$ 4M|E| messages of length J+1

Cost $\sim |E|$

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:

move “bad idea” comment after

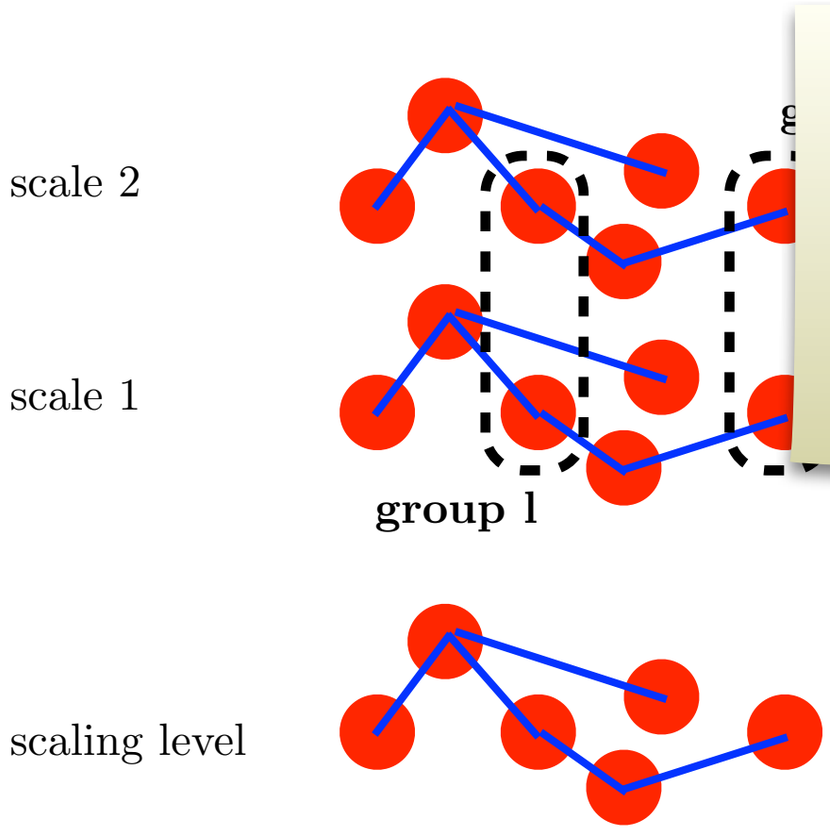
We *know* there are strongly correlated coefficients (LASSO will kill some of them)

There is no information to determine masked wavelets

Group Sparsity - take I

Scaling functions not sparse are optimized separately

Group potentially correlated variables (scales)



put this at the end with
Composite penalties

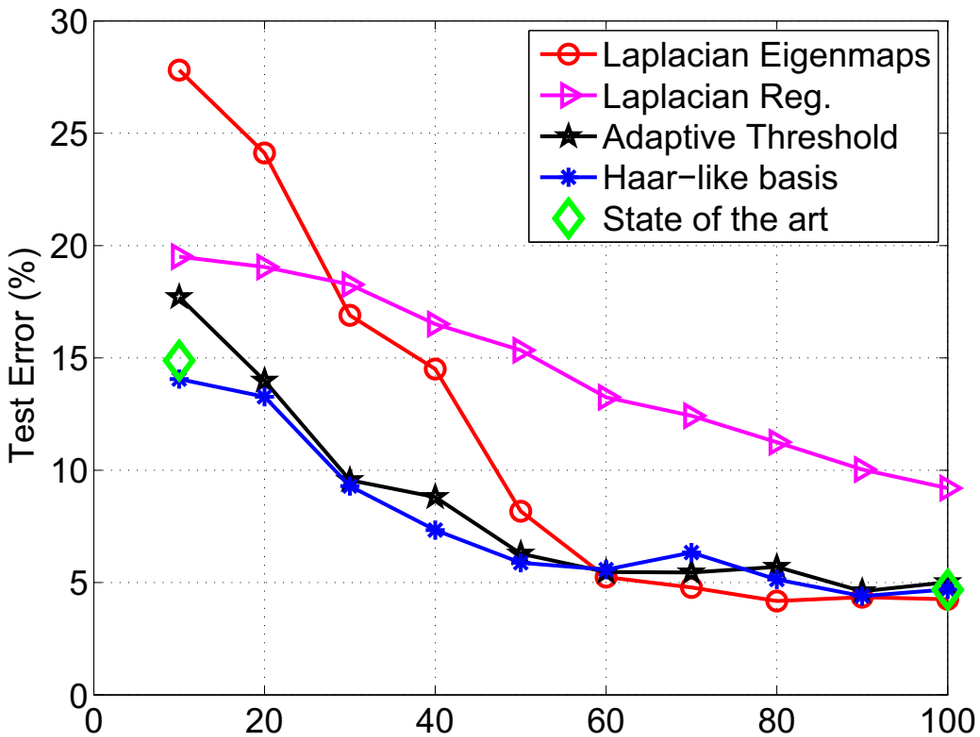
s should be active = local smoothness

Inside group, all coefficients can be active

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

Simple model, no overlap, optimized like LASSO

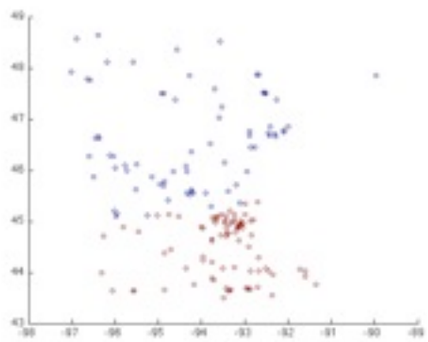
Preliminary Results



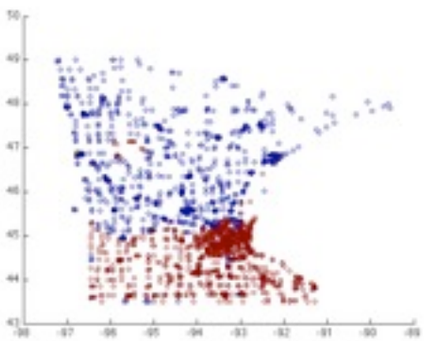
2-class USPS

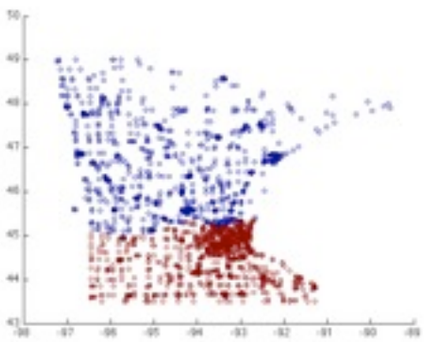
Simulation results from Gavish et al, ICML 2010

5% labeled



recovered





Is it spectacular ?

No. Comparable to state-of-art :(

Group Sparsity - take II (outlook)

Group definition too restrictive

No "spatial" (neighborhood) information

Example (Composite Absolute Penalty [Mosci et al 2010, Jacob, Obozinski, Vert, 2009]):

$$\mathcal{S}(\beta) = \sum_j \gamma_j \sum_{i \in V} \sqrt{\sum_{k \sim i} \beta_{j,k}^2}$$

weights can trigger influence through scales

neighborhood of i

Remarks:

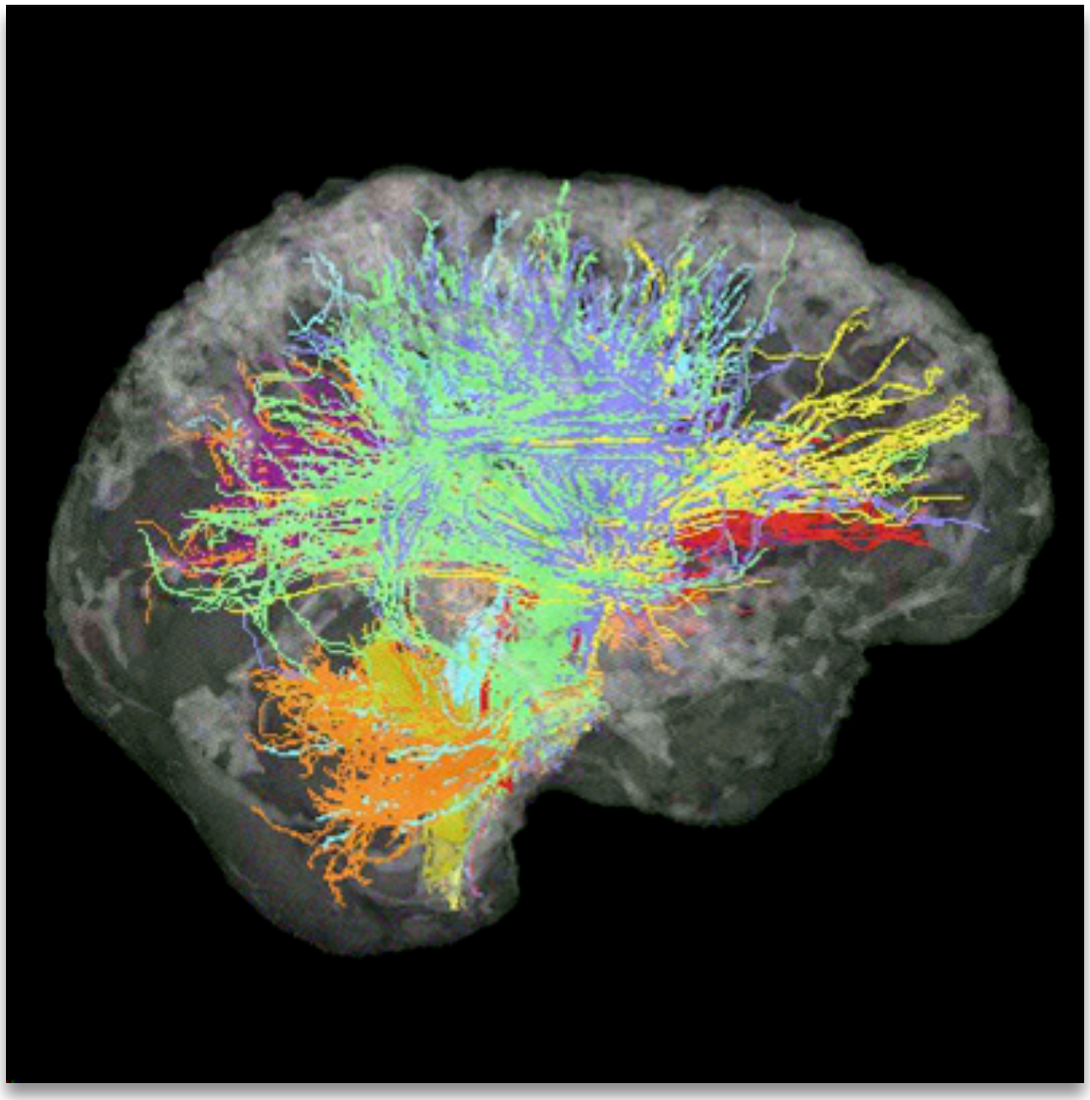
CAP is the composition of mixed norm and adjacency mat.

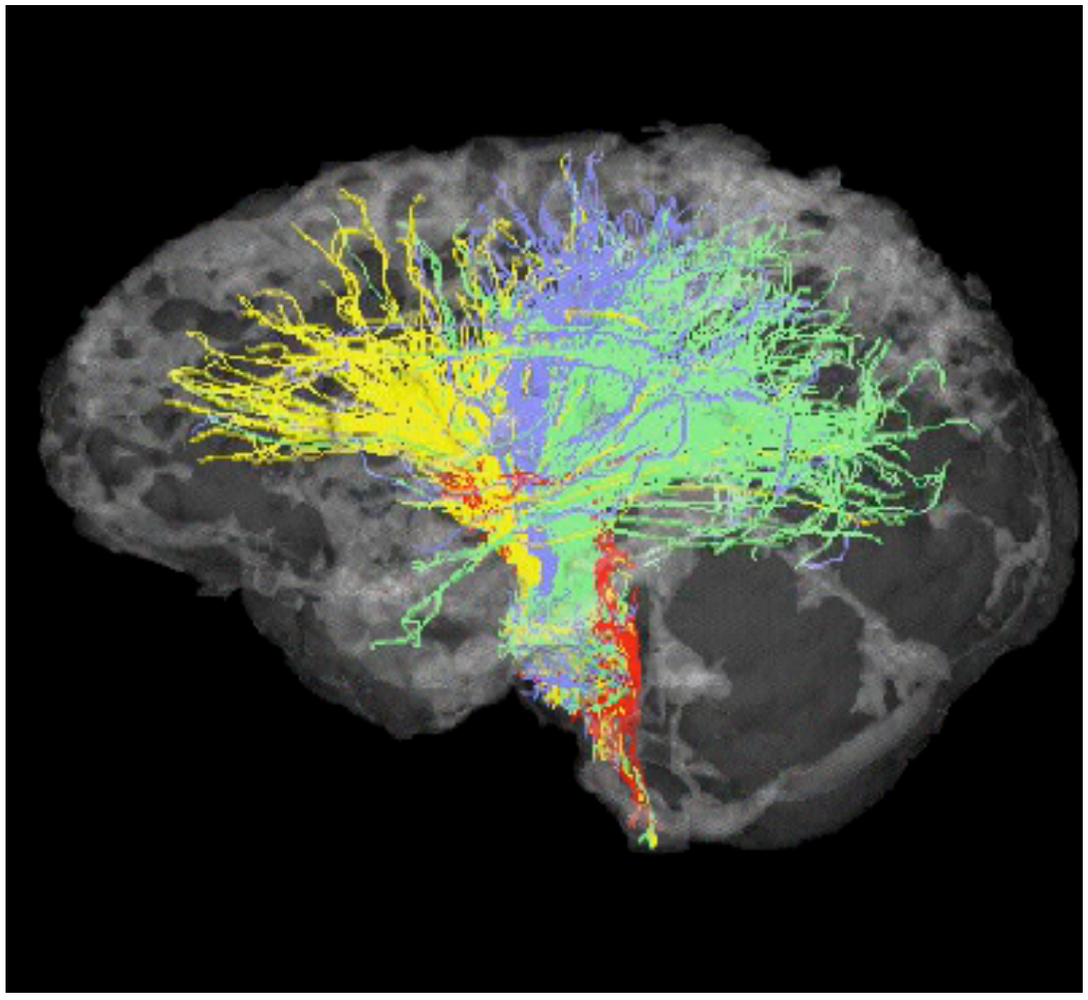
For *analysis* coefficients, at small scale $\sum_{i \in V} \sqrt{\sum_{k \sim i} \beta_{j,k}^2}$ behaves like TV

Conclusions

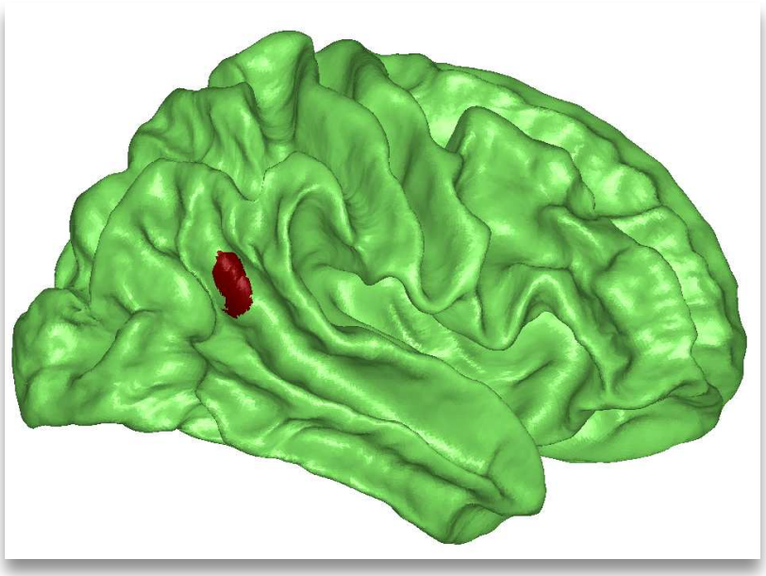
- Structured, data dependent dictionary of wavelets
  - sparsity and smoothness on graph are merged in simple and elegant fashion
  - fast algo, clean problem formulation
  - graph structure can be totally hidden in wavelets
- results not very encouraging for learning
  - on par with state of art but seems more complicated
  - no simple model to cope for information loss
- other applications (sensor net, non-local ...)

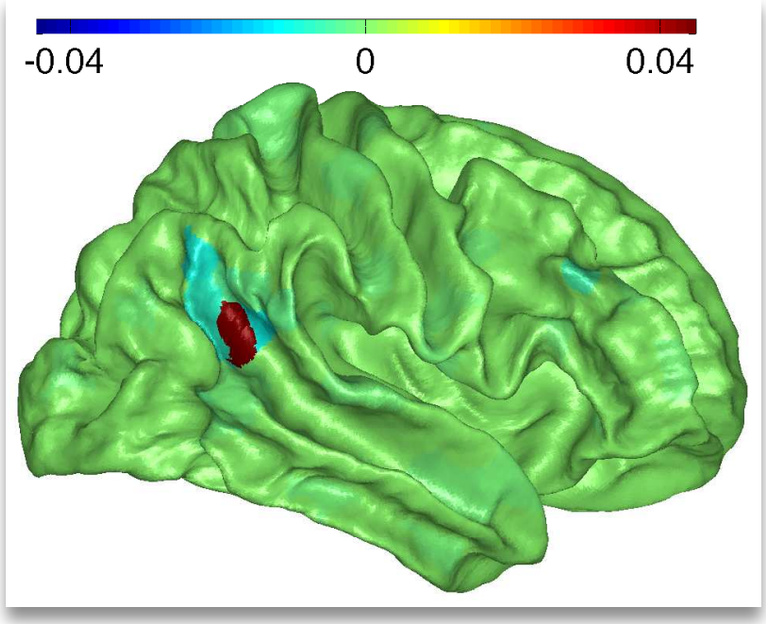
Example

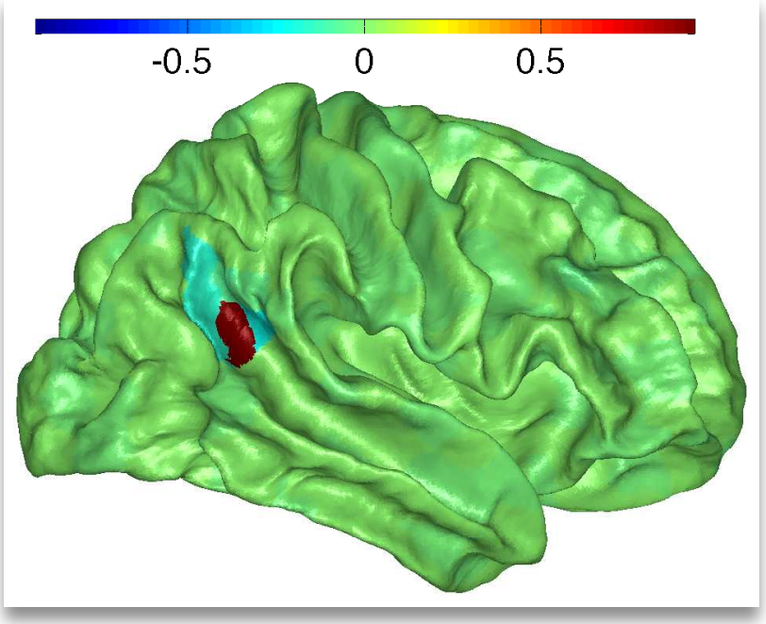


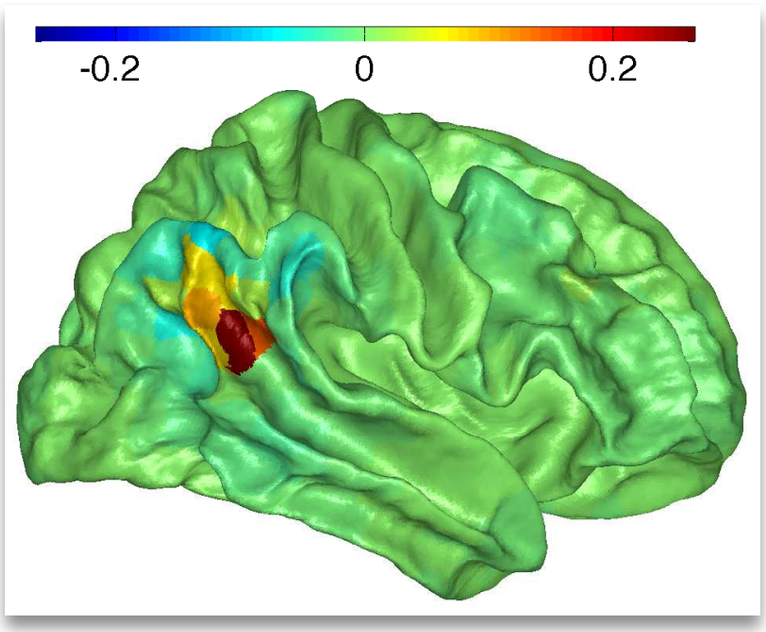


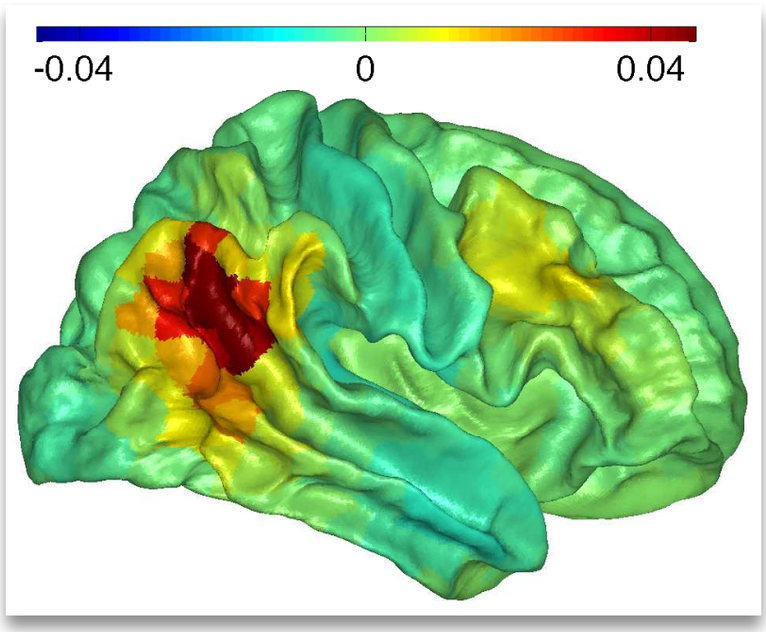
Example

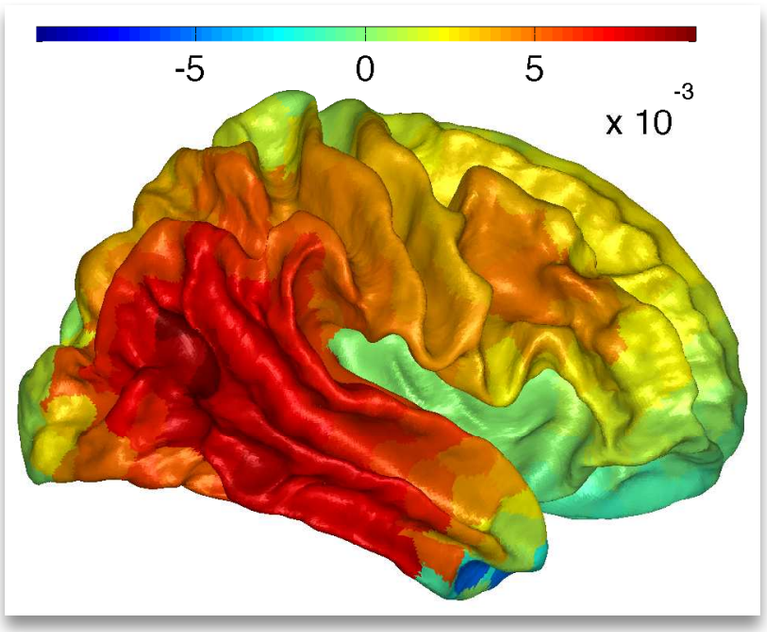






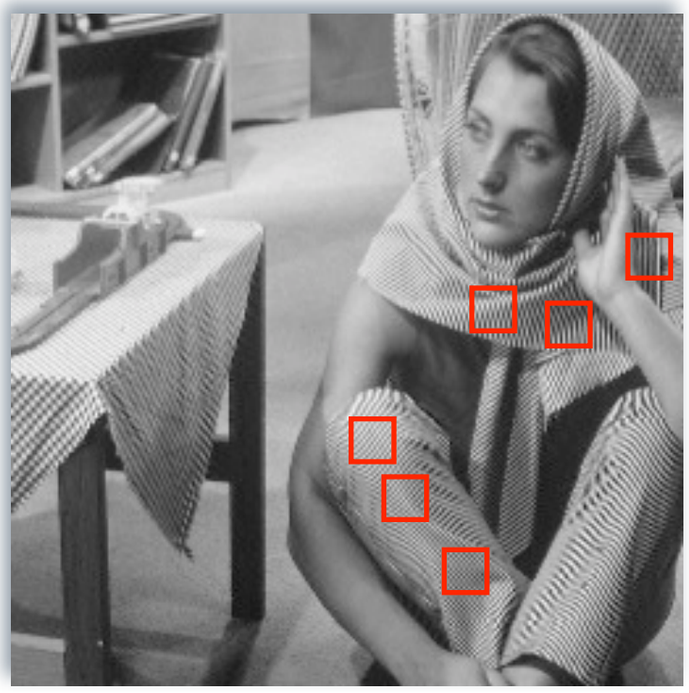


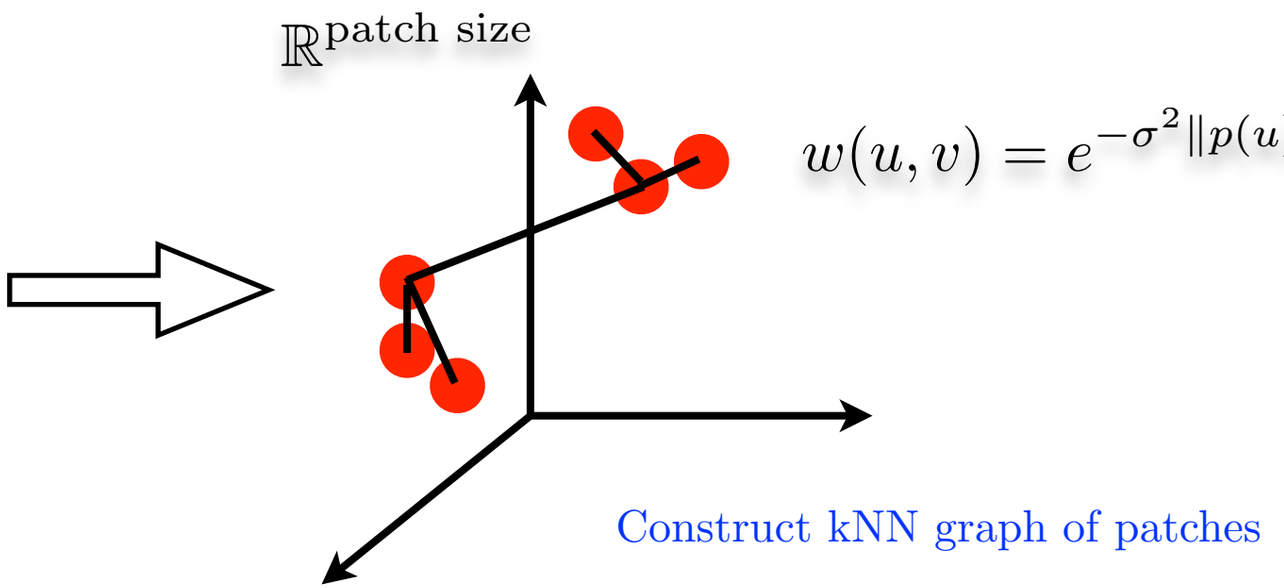




Non-Local Image De-Noising

Non-Local Means, Bilateral Filter: Non-Local Smoothness



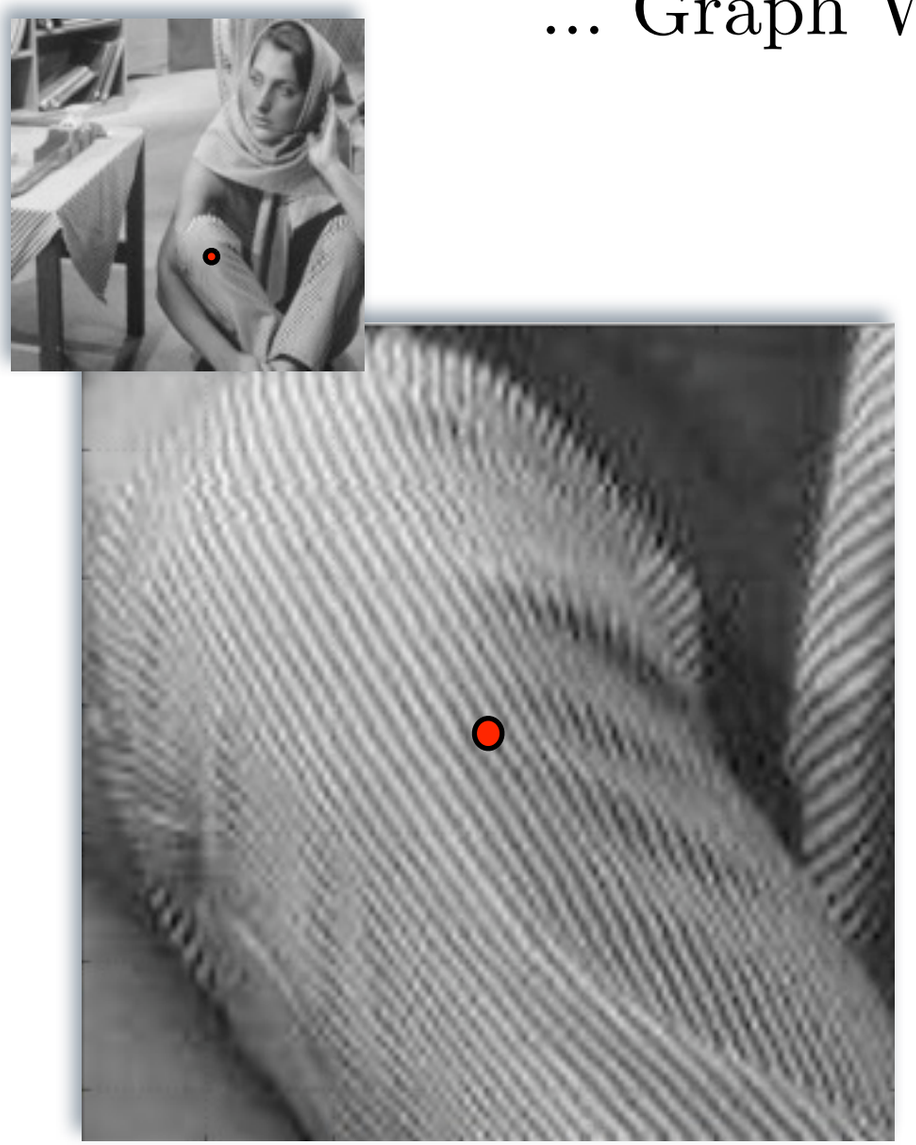


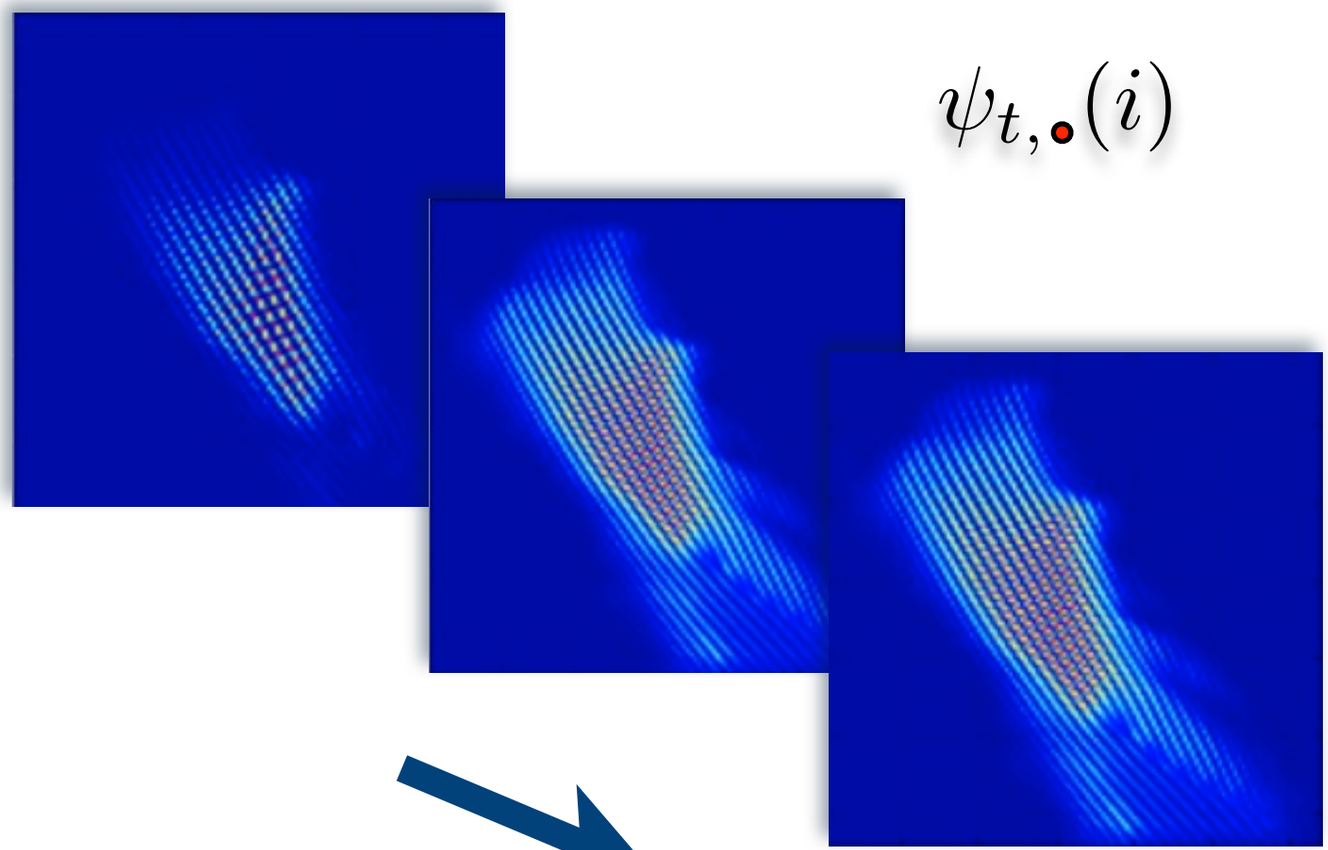
Non-Local Filtering: replace pixel value by weighted average of its *non-local* neighbors $I_{\mathrm{NL}}(u) = \sum_{v \sim u} w(u,v) I(v)$

Non-local Wavelet Frame

- Non-local Wavelets are ...

... Graph Wavelets on Non-Local Graph





increasing scale

Interest: good *adaptive* sparsity basis