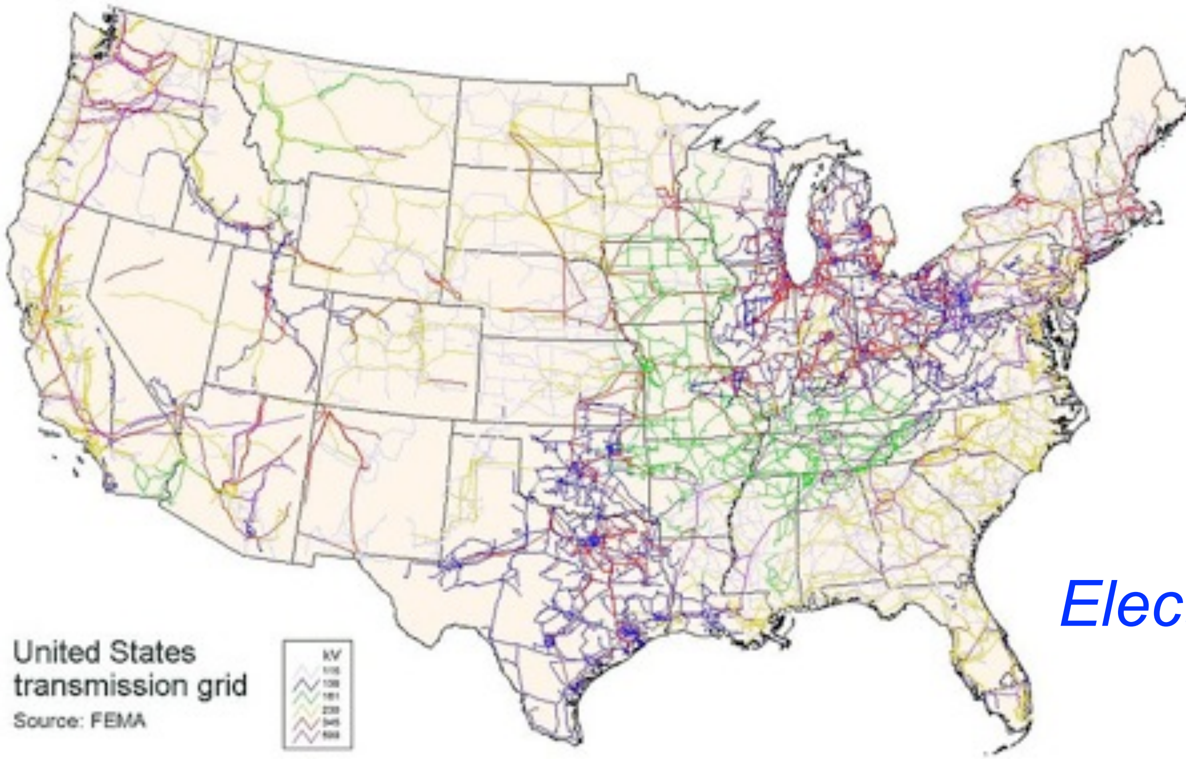
Analyse de données en grande dimension sur graphes et réseaux

Pierre Vandergheynst
Signal Processing Lab, EPFL

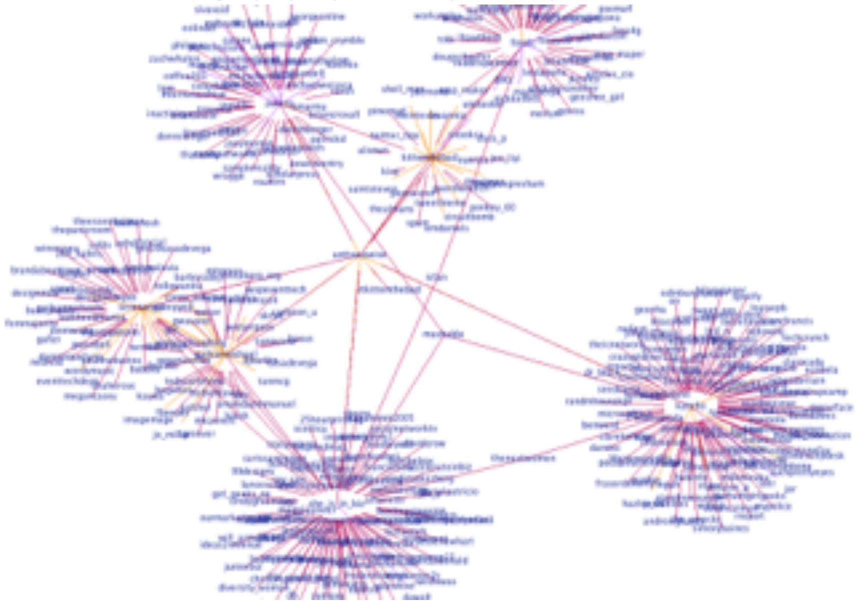
Mathématiques et Grandes Dimensions
Lyon, 10 Décembre 2012

Graphs for Data Modeling



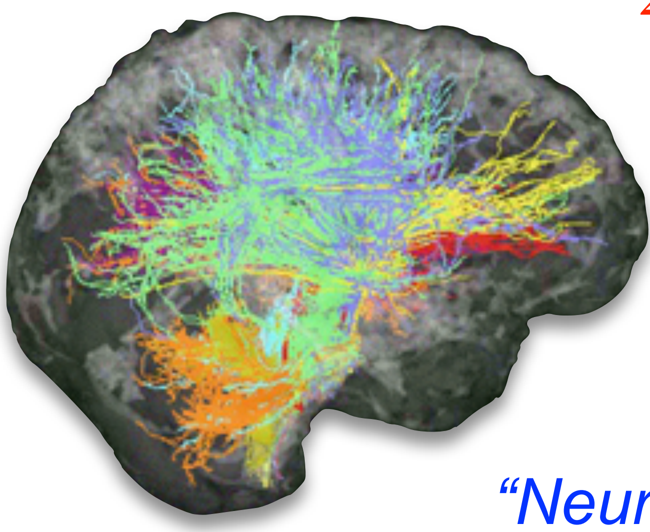
Electrical Network

Social Network



2010: 980 exabytes of new digital information

Big Data



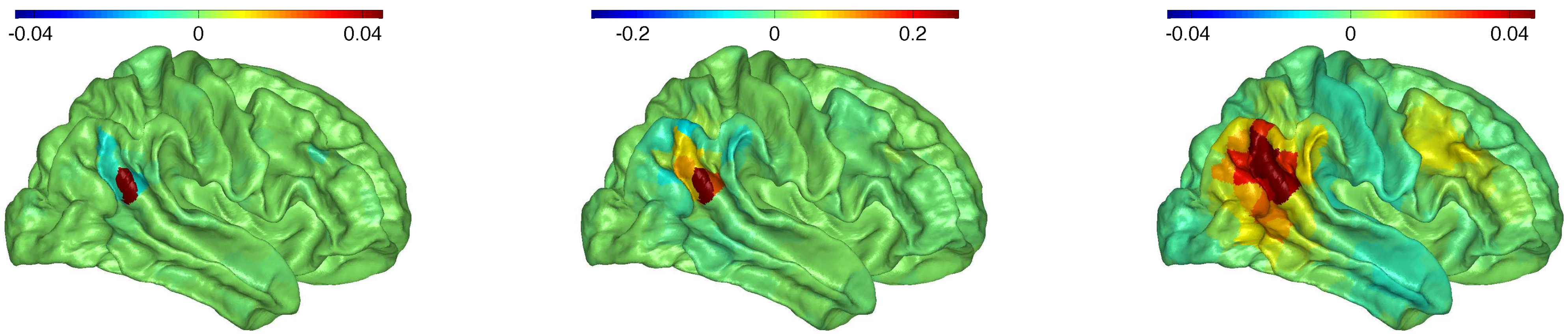
"Neuronal" Network

Ubiquitous sensing

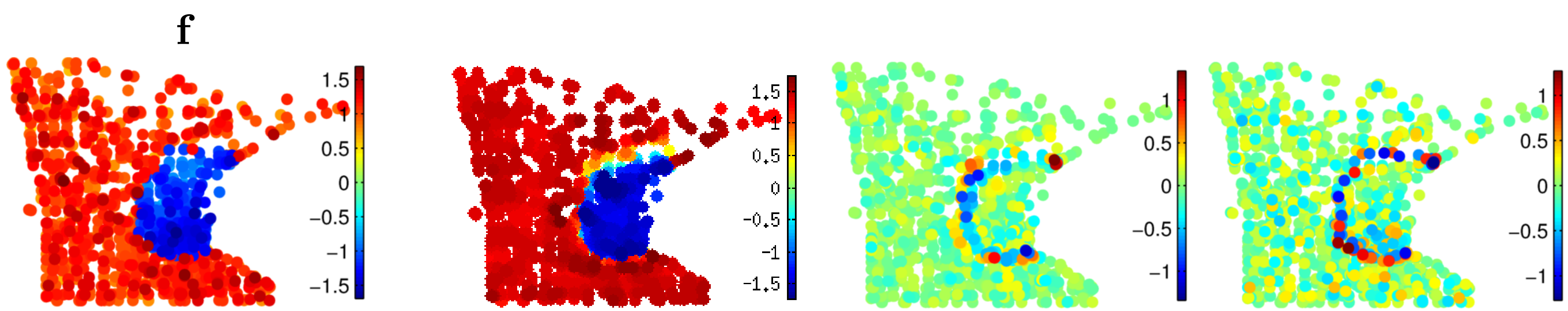
Graphs for Data Modeling

To process or analyze one typically extracts “features” or apply transforms

Local multi-scale averages or multi-scale differentials (wavelets)



The localization/scale properties often induces interesting effects such as sparsity:



Outline

- Wavelets on (undirected) graps
  - Definitions, implementation
  - Localization
- Schematic application to (transductive) learning

Graphs, 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:

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

Graphs, Laplacian and Spectral Theory

eigen decomposition of Laplacian $\Rightarrow$ Spectral Graph Theory

$$\{\chi_l\}_{l=0,1,\ldots,N-1} \qquad 0 = \lambda_0 < \lambda_1 \leq \lambda_2 \ldots \leq \lambda_{N-1} := \lambda_{\max}$$

The Graph Laplacian induces a convenient Fourier-like transform

$$\hat{f}(\ell) := \langle f, \chi_l \rangle = \sum_{n=1}^{N} \chi_\ell^*(n) f(n)$$

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

Smoothness via Laplacian

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"

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

Smoothness via Laplacian

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

Graph Fourier

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

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

"Low pass" filtering !

Simple linear features: $\hat{f}(\ell)\hat{g}(\lambda_\ell; p) \Rightarrow g(\mathcal{L}; p)$

"Convolutions" and "Translations"

$$(f * g)(n) := \sum_{\ell=0}^{N-1} \hat{f}(\ell)\hat{g}(\ell)\chi_\ell(n)$$

Inherits a lot of properties of the usual convolution

associativity, distributivity, diagonalized by GFT

$$g_0(n) \ := \ \textstyle\sum_{\ell=0}^{N-1} \chi_\ell(n) \quad \Longrightarrow \quad f * g_0 = f$$

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

Use convolution to induce translations

$$(T_i f)\,(n) := \sqrt{N}(f * \delta_i)(n) = \sqrt{N}\sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell^*(i)\chi_\ell(n)$$

Spectral Graph Wavelets

$G=(E,V)$ a weighted undirected graph, with Laplacian $\mathcal{L} = D - A$

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

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

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

And so formally define the graph wavelet coefficients of f:

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

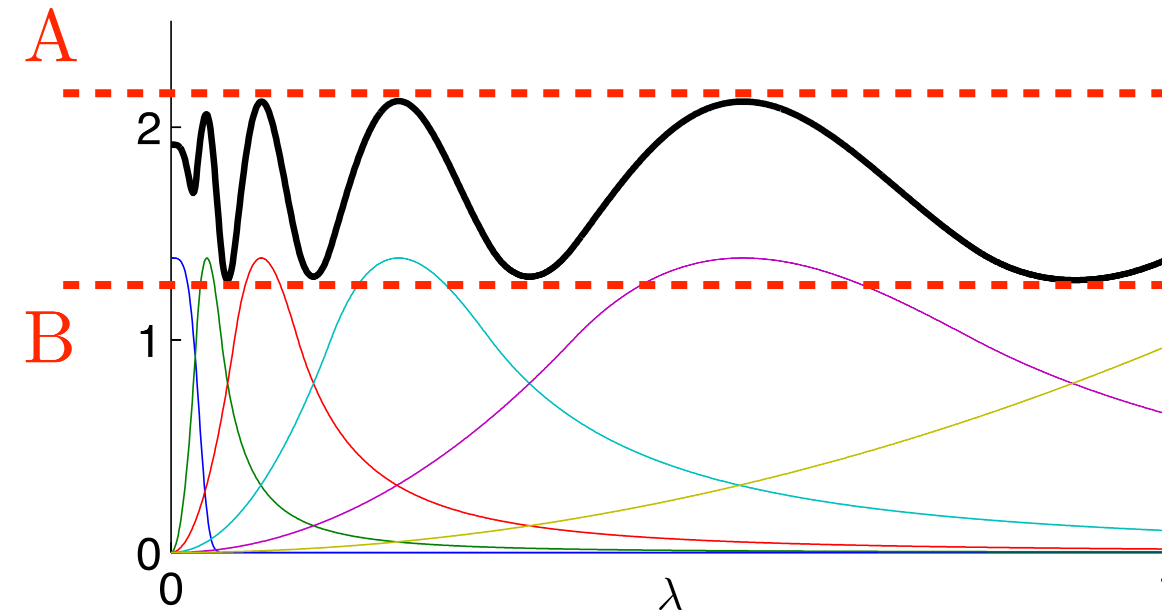
Frames

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

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

scaling function *wavelets*

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

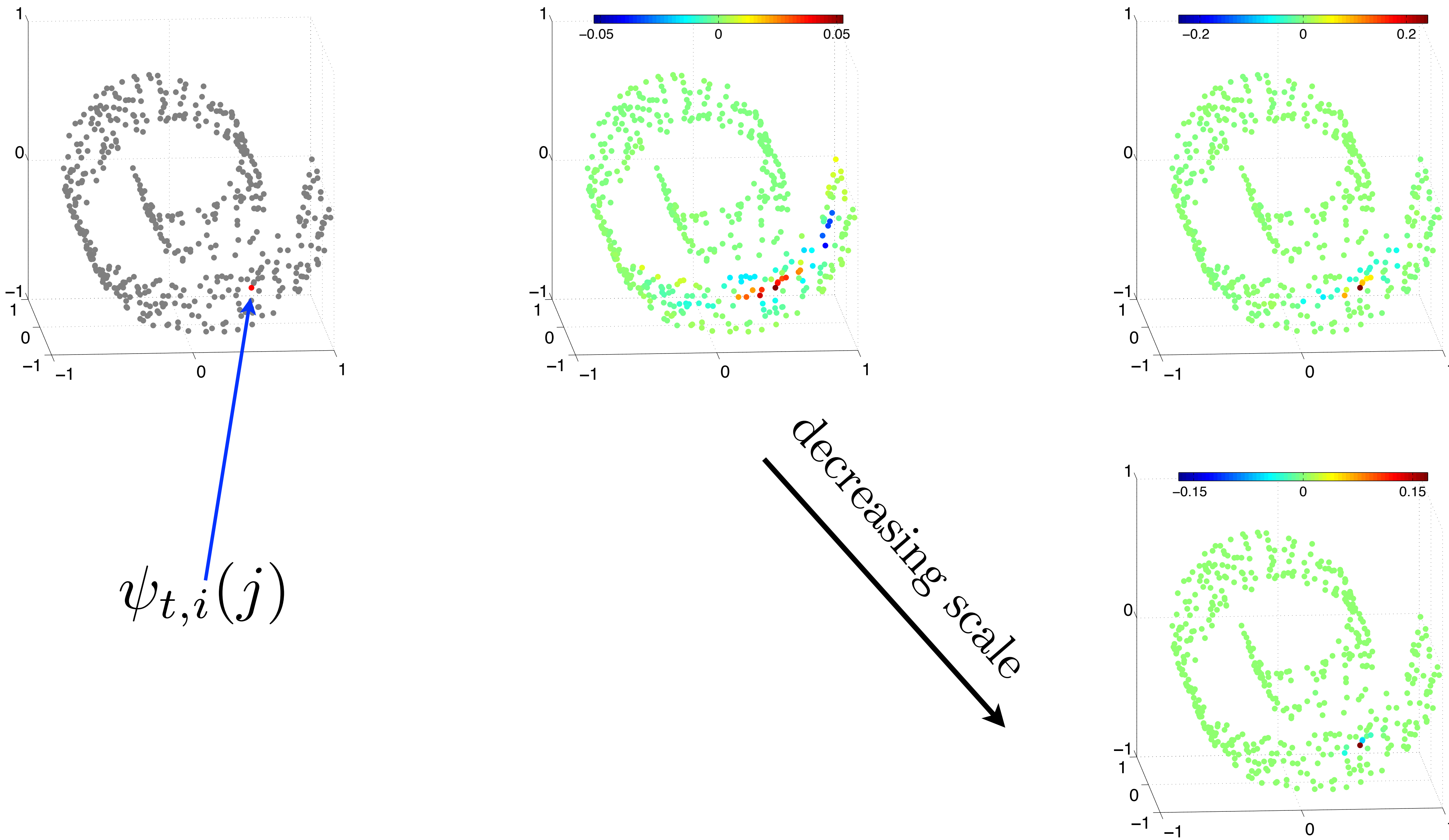


A simple way to get a tight frame:

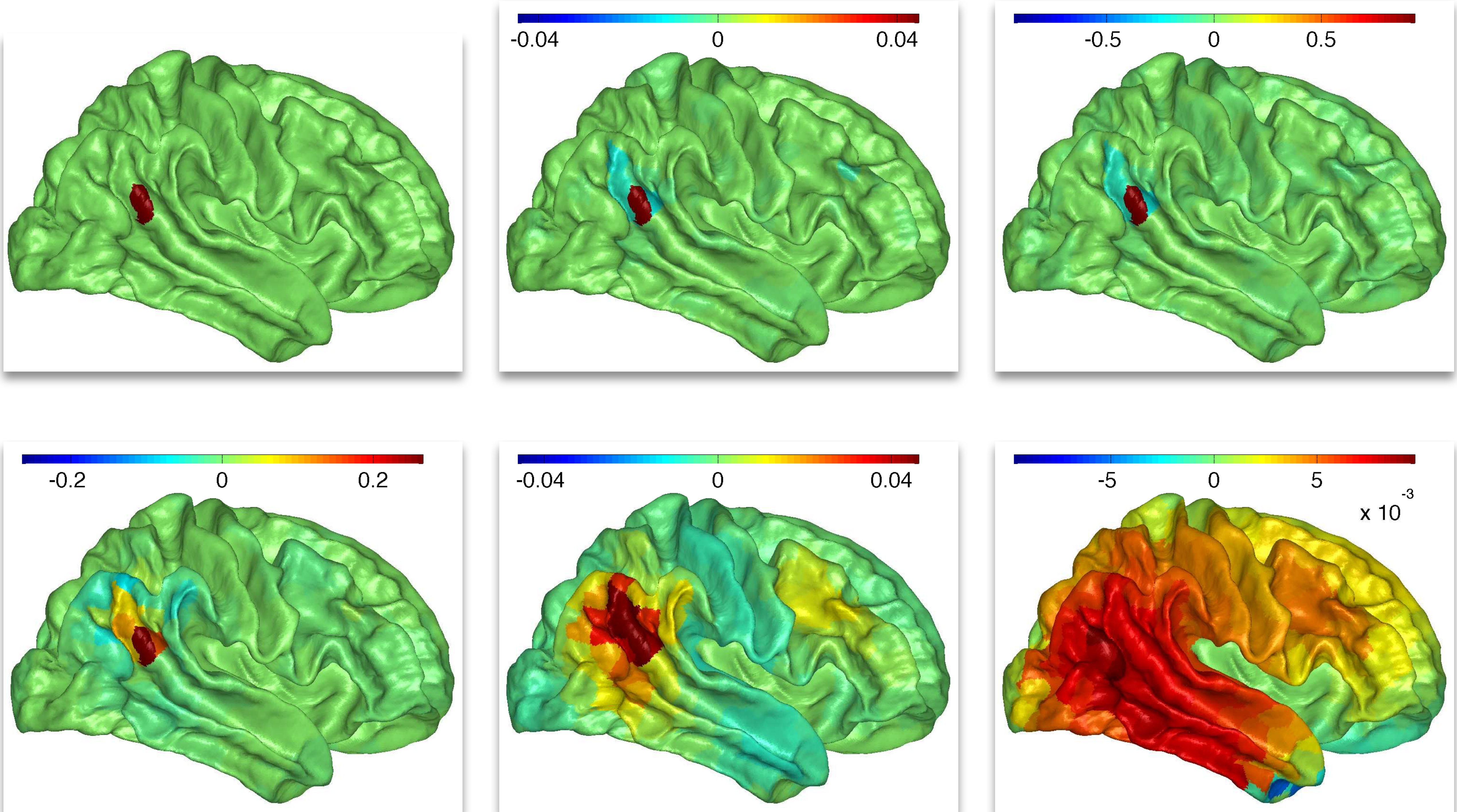
$$\gamma(\lambda_\ell) = \int_{1/2}^{1} \frac{dt}{t}\, g^2(t\lambda_\ell) \implies \tilde{g}(\lambda_\ell) = \sqrt{\gamma(\lambda_\ell) - \gamma(2\lambda_\ell)}$$

for any admissible kernel g

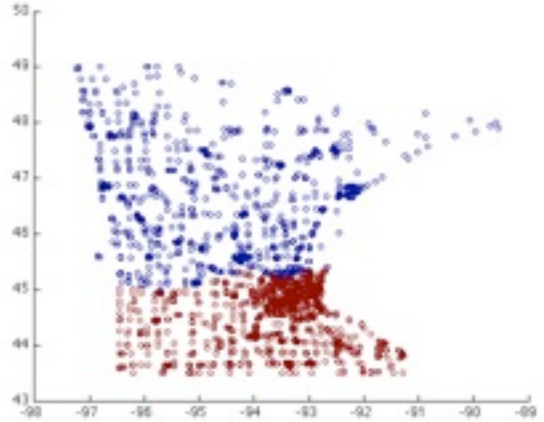
Scaling & Localization



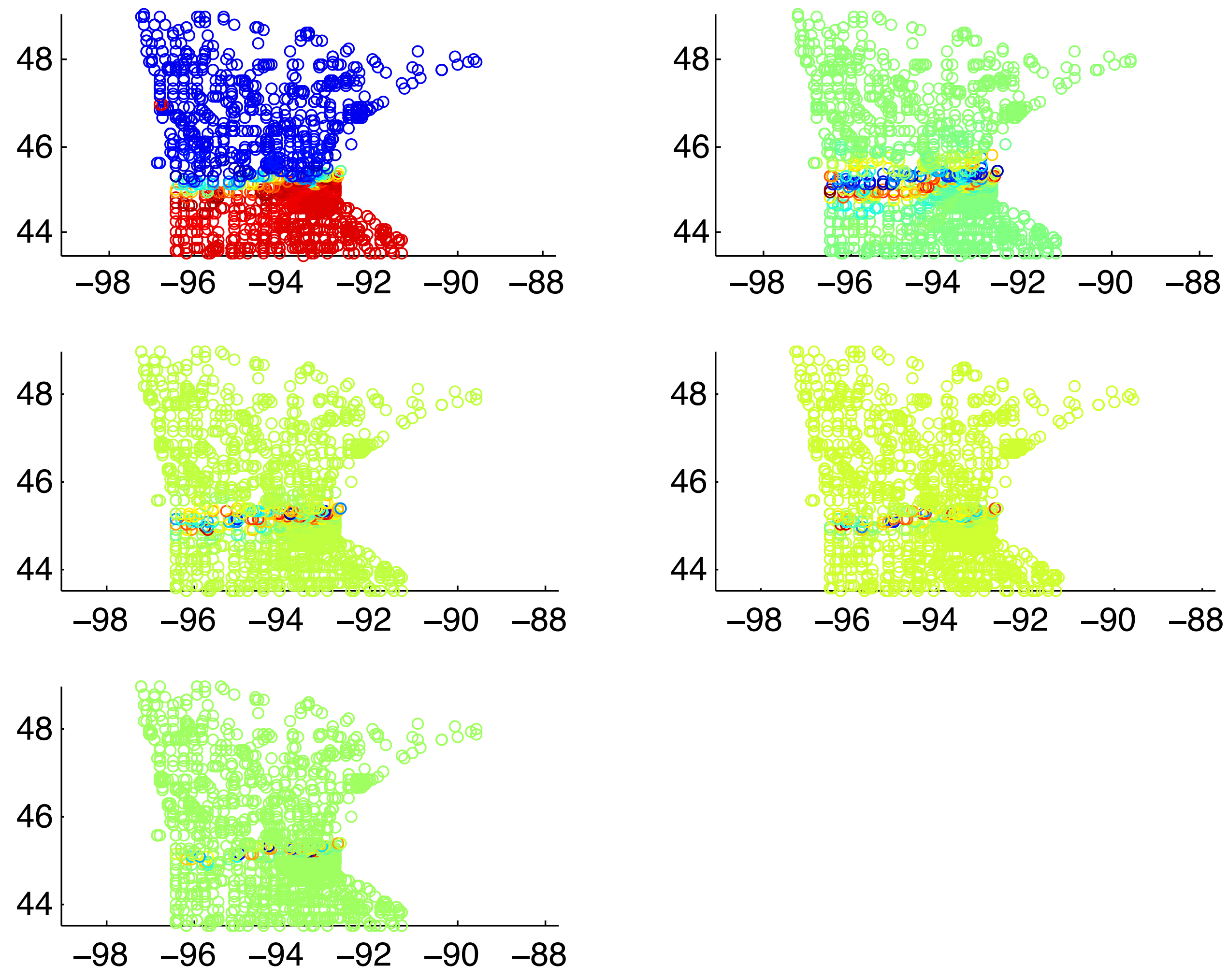
Example



Sparsity and Smoothness on Graphs



scaling functions coeffs



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.):

$$B = \sup_x |\hat{g}^{(K+1)}(x)|$$

Construct an order K polynomial approximation:

$$\sup_\ell |\hat{g}(x) - P_K(x)| \leq \frac{B}{2^K (K+1)!}$$

Polynomial Localization

$$\sup_{\ell} |\hat{g}(x) - P_K(x)| \leq \frac{B}{2^K (K+1)!}$$

Now consider:

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

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

Exactly localized in a K-ball around n

The original feature is well-localized in a K-ball around n:

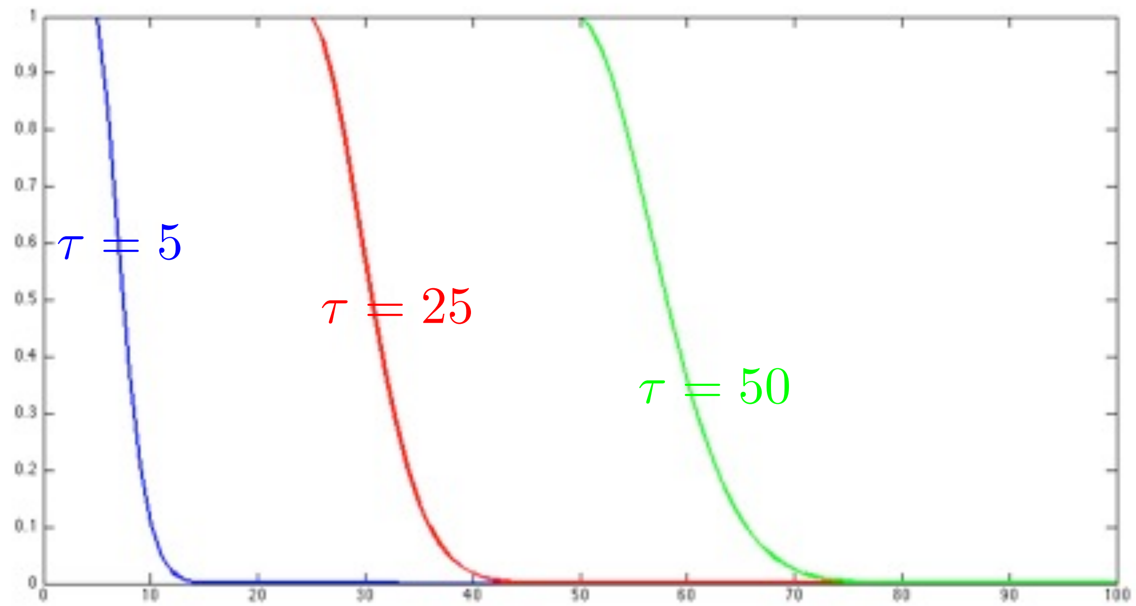
$$B_{\hat{g}}(K) = \inf_{\widehat{p_k}} \left\{ \sup_{\lambda \in [0, \lambda_{\max}]} |\hat{g}(\lambda) - \widehat{p_k}(\lambda)| \right\}$$

$$d_{in} > K \quad |(T_i g)(n)| \leq \sqrt{N} B_{\hat{g}}(d_{in} - 1)$$

Bounds on Localization

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

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



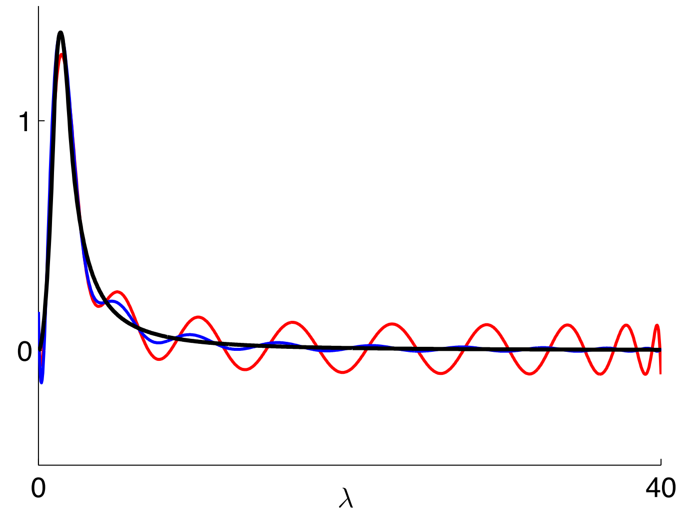
Remark on Implementation

Not necessary to compute spectral decomposition for filtering

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 $O(\sum_{n=1}^{J} M_n|E|)$

http://wiki.epfl.ch/sgwt

Note: “same” algorithm for adjoint !

LTS EPFL LTS2

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

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

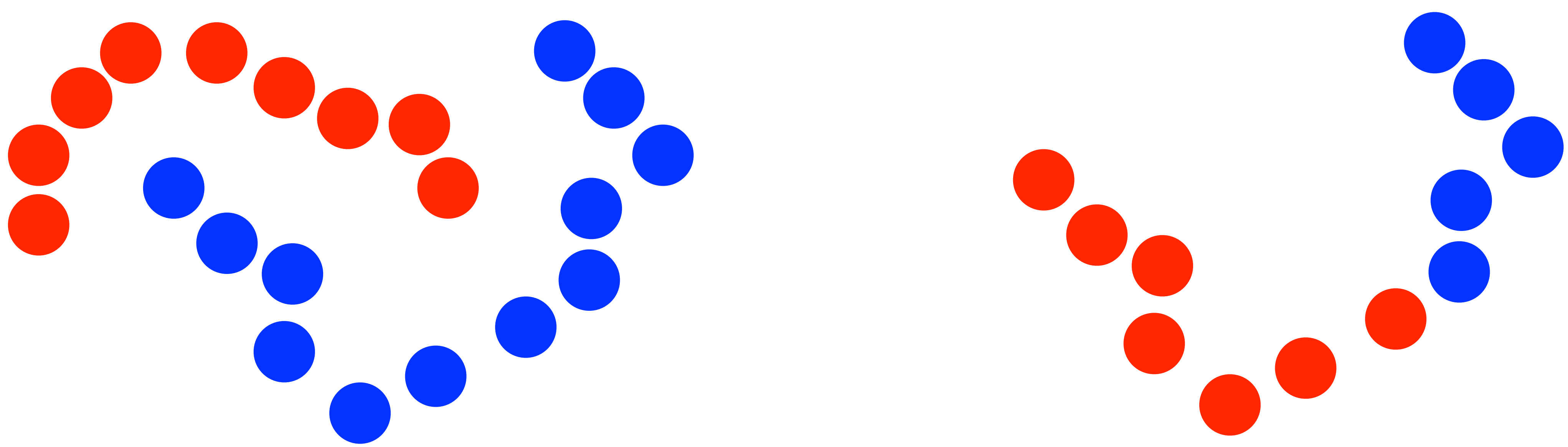
simplifies/stabilizes selected model

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

Transduction & Representation

More general linear model with a dictionary of features ?

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

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

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

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

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 ?

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

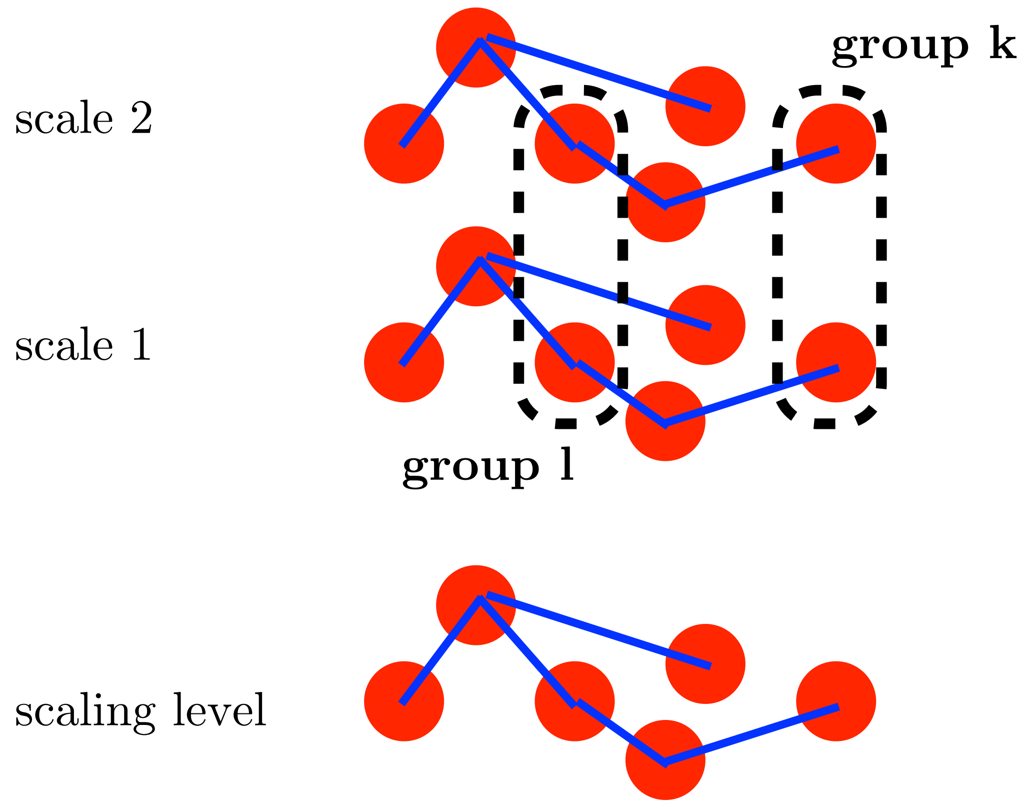
Bad Idea:

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

Group Sparsity

Scaling functions not sparse are optimized separately

Group potentially correlated variables (scales)



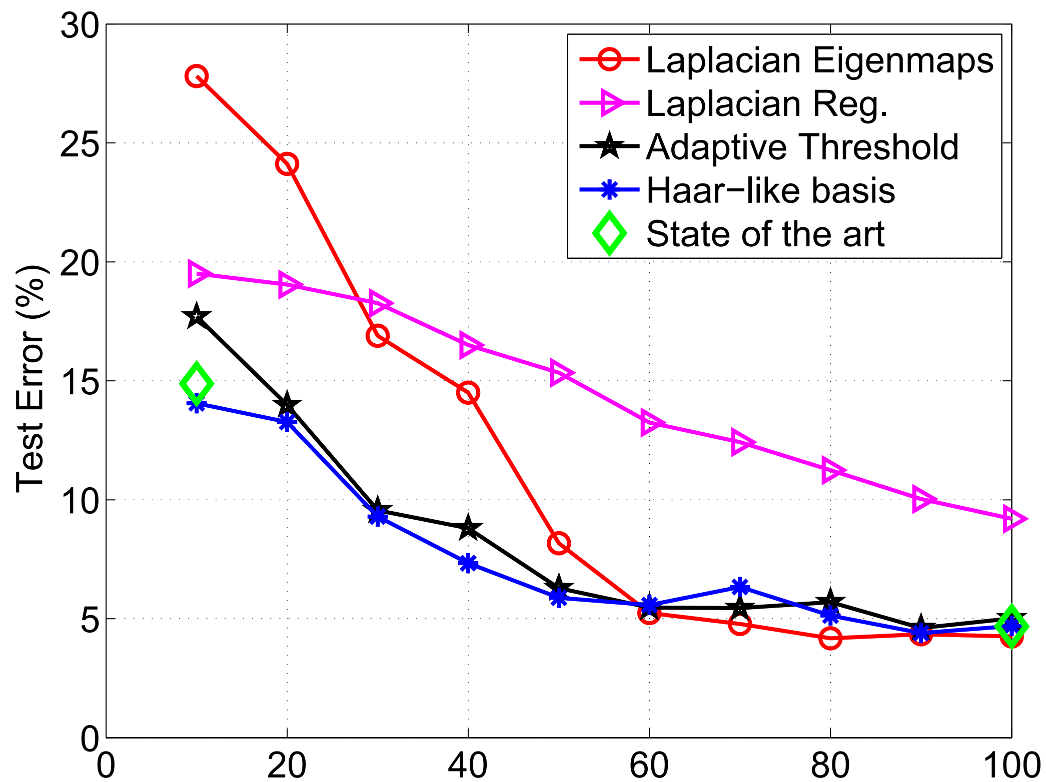
Few groups should be active = local smoothness

Inside group, all coefficients can be active

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

Simple model, no overlap, optimized like LASSO

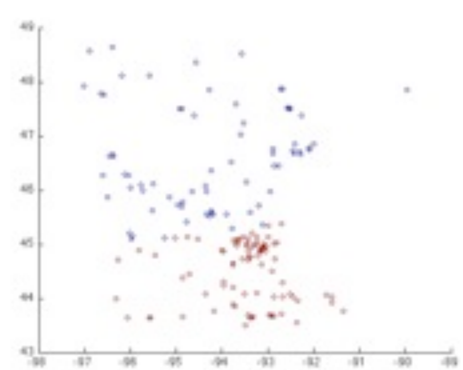
Preliminary Results



2-class USPS

Simulation results from Gavish et al, ICML 2010

5% labeled



recovered

Conclusions

- Processing data on graphs is still an emerging field.
- Interesting connections with other areas
- How to scale computations ?
- Diverse applications:
  - fMRI [Leonardi, Van de Ville, 2012], cortical smoothing
  - Network Analysis [Tremblay, Borgnat, 2012]
  - Learning, Distributed regularization [Shuman et al, 2012]