Data processing and networks: Basics in Graph Signal Processing

The question for this lecture: how to mimic Signal Processing for data on graphs ?

Hence, in 1 slide: **what is Signal processing ?**

$\boxed{\text{Information}} \xrightarrow{\text{Coding}} \boxed{\text{Data}} \xrightarrow{\text{Measuring}} \boxed{\text{Observations}}$

Noise/Missing

e.g.:

Information	Data	Observations
× Climate change?	Temperature (t)	Samples at specific times/places
× Gene Expression profiling?	DNA microarray	Images of spots.
× Music (or Speech)	Sound	Noisy sound, sampled, ...

Need to define what is information, how it affects the data, what is interesting, ...

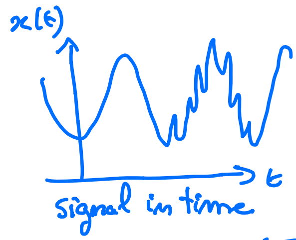
Need a model of measurement and observation

The way back = Signal Processing!

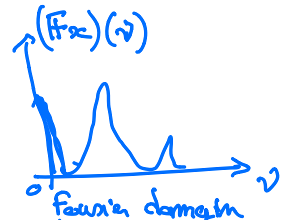
Key lessons from Signal processing:

- **Representation of data is important**

$x(t)$ or $(\mathcal{F}x)(\nu)$ or $x(t,\nu)$ or ...



or



$$(\mathcal{F}x)(\nu) = \int_{-\infty}^{+\infty} x(t)\, e^{-i2\pi\nu t}\, dt$$

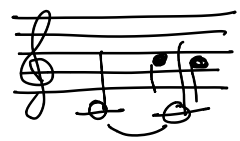
- **Know how to write observation models**

Observation, signal

$$o(t) = \underbrace{x(t-\tau)}_{\text{signal}} + n(t)$$

τ: delay; $n(t)$: noise

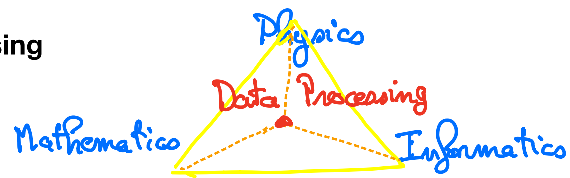
or



?

- **Two types of tools are required:**
    - **Exploratory data analysis (know how to better display information)**
    - **Exact tools for inference (know to best extract information, with statistical confidence)**

. The golden triangle of Signal processing



Harmonic analysis on graphs

1) Definitions and notations

A graph is a pair with a set of vertices (or nodes) V and a set of edges (or links) E

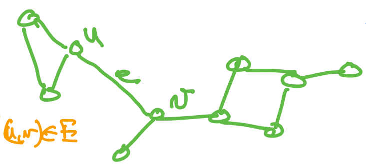
$G=(V,E)$ where $e=(u,v)\in E$ codes for an edge between $u\in V$ and $v\in V$

Weighted graph:

$G=(V,E,W)$

with $W: V\times V \to \mathbb{R}$

$(u,v) \mapsto w(u,v) \neq 0$ iff $(u,v)\in E$



Note: $N = \text{Card}\, V = |V|$

$M = \text{Card}\, E$

Many properties can be studied through linear algebra

- Adjacency matrix: $\underline{\underline{A}}$ such that $(\underline{\underline{A}})_{uv} = \begin{cases} 1 & \text{if } (u,v)\in E \\ 0 & \text{otherwise} \end{cases}$

  or w if weighted.

- Degree of a node: $d(u) = \#$ of edges having u as origin/destination

  $\underline{\underline{D}} = \text{diag}(d(u), d(v), \dots)$ $\quad = \sum_{v\in V} A_{uv} = (\underline{\underline{A}}\cdot \underline{1})_u$

- Random walk on a graph: with $\underline{\underline{P}} = \underline{\underline{D}}^{-1}\underline{\underline{A}}$, the position X_t of a walker satisfies: $\mathbb{P}(X_{t+1}=u) = \sum_v \mathbb{P}(X_t=v)\, P_{vu}$.

  (probability that the random walker is at node u at time $t+1$)

2) Regularity and the Laplacian operator

Let us consider a function (some data) on the graph : $f : \begin{cases} V \longrightarrow \mathbb{R} \ (\text{or } \mathbb{C}, \text{ or } \mathbb{R}^d, \text{ or} \dots) \\ u \longmapsto f(u) \quad (\text{or } f_u) \end{cases}$

remark : if $\text{Card } V = |V| < +\infty$,
then the image of V through f is a vector $\underline{f}$ s.t. $\underline{f}_u = f(u)$

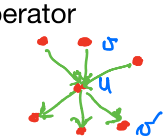
We want to define the derivatives (or gradients) on the graph

$\forall (u,v) \in V \times V \quad \nabla f(u,v) = \sqrt{A_{uv}} \, (f(v) - f(u))$ RK $\sqrt{A_{uv}}$ for weights !

The operator gradient maps function on V to objects on E

Then, we could have a complete Discrete calculus on graphs, e.g. a divergence operator

For $g : E \longrightarrow \mathbb{R}$, $(\text{div } g)(u) = \sum_{v \,|\, (u,v) \in E} \sqrt{A_{uv}} \, g(u,v) - \sum_{v' \,|\, (v',u) \in E} \sqrt{A_{v'u}} \, g(v',u)$



In matrix form : $\text{div} \equiv \underline{\underline{S}}$ where

$$\underline{\underline{S}} = \begin{array}{c} e=(u,v) \\ \begin{pmatrix} & 0 & \\ & -1 & \\ & 0 & \\ & +1 & \\ & 0 & \end{pmatrix} \end{array} \begin{array}{c} \\ \\ u \\ \\ v \\ \\ \end{array} \in \mathbb{R}^{|V| \times |E|}$$

$\underline{\underline{S}}$ is called the incidence matrix,

Also we can remark that $\boxed{\text{div} \equiv \nabla^T \equiv \underline{\underline{S}}}$ hence $\boxed{\nabla \equiv \underline{\underline{S}}^T}$

Definition: the **Laplacian operator** of a undirected graph is defined as

$$\underline{\underline{L}} = \underline{\underline{D}} - \underline{\underline{A}} \qquad \text{hence} \quad (\underline{\underline{L}})_{uu} = d_u \ \text{the degree}$$

$$\text{for } u \neq v \quad (\underline{\underline{L}})_{uv} = -1 \ \text{if } (u,v) \text{ is an edge}$$
$$= 0 \ \text{otherwise}$$

Propr: • $(\underline{\underline{L}}\,\underline{f})_u = d_u f_u - \sum_{v \in V \setminus \{u\}} A_{uv} f_v$

$$= \sum_{v \in V} A_{uv} (f_u - f_v) \qquad \text{because } d_u = \sum_{v \in V} A_{uv}$$

$$= (\underline{\underline{S}}\,\underline{\underline{S}}^T \underline{f})_u \Longrightarrow \boxed{\underline{\underline{L}} = \underline{\underline{S}}\,\underline{\underline{S}}^T}$$

Said differently: Laplacian = divergence of gradient.

• Let us introduce the scalar product between two functions

$$\langle \underline{f}, \underline{g} \rangle = \sum_{v \in V} f(v)\, g(v) = \underline{f}^T \underline{g}$$

then

$$\langle \underline{f}, \underline{\underline{L}}\,\underline{f} \rangle = \underline{f}^T \cdot \underline{\underline{L}}\,\underline{f}$$
$$= \frac{1}{2} \sum_{u,v \,|\, (u,v) \in E} A_{uv} \big(f(u) - f(v)\big)^2 \geq 0$$

This is called the Dirichlet form and it measures the global variations of the function f on G.

Remarks: 1) Several definitions for directed graphs (not detailed here)

2) Normalized Laplacian:

$$\underline{\underline{L}}_n \hat{=} \underline{\underline{D}}^{-1/2} \underline{\underline{L}}\, \underline{\underline{D}}^{-1/2} = \underline{\underline{Id}} - \underline{\underline{D}}^{-1/2} \underline{\underline{A}}\, \underline{\underline{D}}^{-1/2}$$

It works the same way with

$$\langle \underline{f}, \underline{\underline{L}}_n \underline{f} \rangle = \frac{1}{2} \sum_{(u,v) \in E} A_{uv} \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}} \right)^2$$

Spectral analysis on graphs:

Thanks to the Laplacian, one can define **a spectral domain and a Fourier transform on G**

- Prop. of $\underline{\underline{L}}$: $\underline{\underline{L}}$ is symmetric and positive semi-definite $\left(\forall \underline{f},\ \langle \underline{f}, \underline{\underline{L}}\,\underline{f}\rangle \geq 0\right)$

  Csq. : $\underline{\underline{L}}$ is diagonalizable with real and non-negative eigenvalues.

  Because if $\underline{\chi_k}$ is an eigenvector of $\underline{\underline{L}}$ with eigenvalue λ_k,

  then $\underbrace{\langle \underline{\chi_k}, \underline{\underline{L}}\,\underline{\chi_k}\rangle}_{\geq 0} = \langle \underline{\chi_k}, \lambda_k \underline{\chi_k}\rangle = \lambda_k \underbrace{\langle \underline{\chi_k}, \underline{\chi_k}\rangle}_{\geq 0}$ $\Rightarrow \lambda_k \geq 0$

  Writing $\underline{\underline{\Lambda}} = \mathrm{diag}(\lambda_0, \dots \lambda_{N-1})$ and $\underline{\underline{\chi}} = \left(\underline{\chi_0} \middle| \underline{\chi_1} \middle| \dots \middle| \underline{\chi_{N-1}}\right)$,

  we have $\boxed{\underline{\underline{L}} = \underline{\underline{\chi}}\,\underline{\underline{\Lambda}}\,\underline{\underline{\chi}}^T}$

- Spectral domain : the basis of eigenvectors of $\underline{\underline{L}}$ is called its spectral domain.

  By analogy with usual spectral domains, one defines the

  Graph Fourier Transform : $\boxed{(\mathbb{F}_G f)(\lambda_k) \hat{=} \langle \underline{\chi_k}, \underline{f}\rangle} = \underline{\chi_k}^T \cdot \underline{f}$
  $= \sum_v \chi_k(v) f(v)$

  More simply, this GFT is : $\boxed{\underline{\tilde{f}} = \underline{\underline{\chi}}^T \cdot \underline{f}}$

• Prop of the GFT : i) it is invertible $\quad \underline{f} = \underline{\underline{\chi}}\,\underline{\tilde{f}}$ because $\underline{\underline{\chi}}\,\underline{\underline{\chi}}^T = \underline{\underline{Id}}$

ii) Theorem of Parseval: $\langle f, g\rangle = \langle \tilde{f}, \tilde{g}\rangle$

• Prop of the eigenvectors of $\underline{\underline{L}}$:

× $\underline{\underline{L}}$ always admits 0 as an eigenvalue, because $\underline{\underline{L}} \cdot \underline{1} = 0$

× the multiplicity of eigenvalue 0 is equal to the # of connected components in G

× the eigenvector associated to the smallest, non zero, eigenvalue is called the Fiedler vector. Let us note them λ_1 and $\underline{\chi_1}$.

Then $\underline{\chi_1}$ is a crude oscillation (because $\langle \chi_1, 1\rangle = 0$) and it is the smoothest possible, solution of $\arg\min_{\underline{f}/\|\underline{f}\|=1} \langle \underline{f}, \underline{\underline{L}}\,\underline{f}\rangle$

↳ hence of the lowest frequency

× Next eigenvectors: $\underline{\chi_k} = \arg\min_{\underline{f} \in Span(\chi_0, \dots, \chi_{k-1})} \dfrac{\langle \underline{f}, \underline{\underline{L}}\,\underline{f}\rangle}{\langle \underline{f}, \underline{f}\rangle}$.

↳ always the remaining oscillation at lowest frequency

• A Fundamental Analogy: $\left\{ \begin{array}{l} \underline{\chi_k} \longleftrightarrow \text{Fourier mode, oscillation} \\ \lambda_k \longleftrightarrow \text{(squared) Frequencies.} \end{array} \right.$

A simple example: the straight line

•

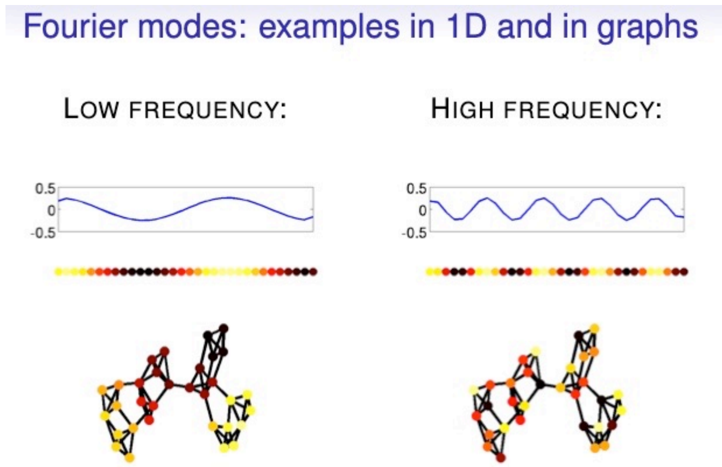
$\longleftrightarrow \quad L = \begin{pmatrix} \dots & & & & & & \\ \dots & -1 & 0 & 0 & 0 & 0 & \\ \dots & 2 & -1 & 0 & 0 & 0 & \\ & -1 & 2 & -1 & 0 & 0 & \\ & 0 & -1 & 2 & -1 & 0 & \\ & 0 & 0 & -1 & 2 & -1 & \\ & 0 & 0 & 0 & -1 & 2 & \dots \\ & 0 & 0 & 0 & 0 & -1 & \dots \\ & & & & & & \dots \end{pmatrix}$

For this regular line graph, L is the 1-D classical laplacian operator (i.e. double derivative operator):

its eigenvectors are the Fourier vectors, and its eigenvalues the associated (squared) frequencies

Examples of Fourier modes ; oscillation and smoothness

[Tremblay, PB]



[Tremblay, Gonçalves, PB, 2017]

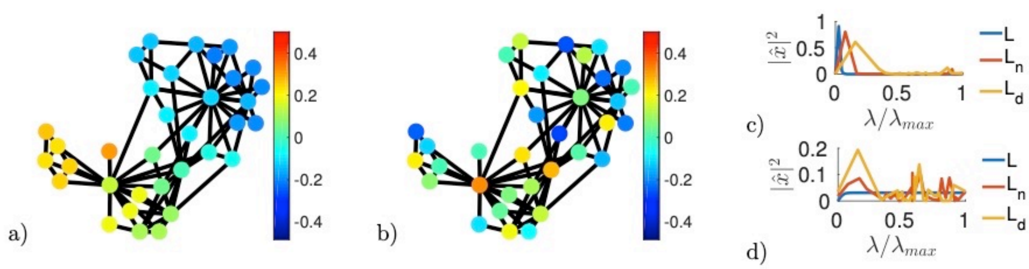
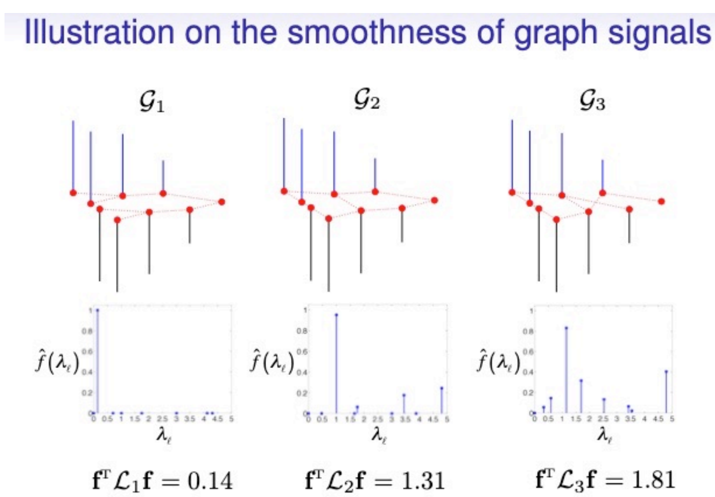


Figure 1: ***Two graph signals and their GFTs.*** *Plots a) and b) represent respectively, a low-frequency and a high-frequency graph signal on the binary Karate club graph [21]. Plots c) and d) are their corresponding GFTs computed for three reference operators:* $\mathbf{L}$, $\mathbf{L_n}$ *and* $\mathbf{L_d}$ *(equivalent to the GFT defined via the adjacency matrix).*

[Vandergheynst & Shuman, 2013]



Functional calculus on graph

Objective: define the effect of function on graph data

We use the simple property that $\underline{L}^n \underline{\chi_k} = \lambda_k^n \underline{\chi_k}$

Then, for any polynomial function f, we have
$$f(\underline{L}) = \sum_{\lambda_k \in Sp(\underline{L})} f(\lambda_k)\, \underline{\chi_k}\, \underline{\chi_k}^T = \underline{X}\, f(\underline{\Delta})\, \underline{X}^T$$

Using approximation theorem, it holds for any function.

Example: define a **diffusive** process on a graph

With the analogy: $f(u,t)$ is a diffusion if it follows $\boxed{\frac{\partial f}{\partial t} = -\underline{L}\, f}$

Applying the GFT: $\frac{\partial}{\partial t} \tilde{f}(\lambda_k, t) = -\lambda_k \tilde{f}(\lambda_k, t)$

hence, if $f(u,t=0) = f_0(u)$, we have $\tilde{f}(\lambda_k, t) = e^{-t\lambda_k} \tilde{f}_0(\lambda_k)$

With fct^{al} calculus: $\boxed{\underline{f}(t) = e^{-t\underline{L}}\, \underline{f_0}}$

Explicit expression: $f(u,t) = \sum_k \underbrace{e^{-t\lambda_k} \tilde{f}_0(\lambda_k)} \chi_k(u)$

This acts as a filter $e^{-t\lambda_k}$ on the GFT of the initial condition $\underline{f_0}$.

478 B. Ricaud et al. / C. R. Physique 20 (2019) 474–488

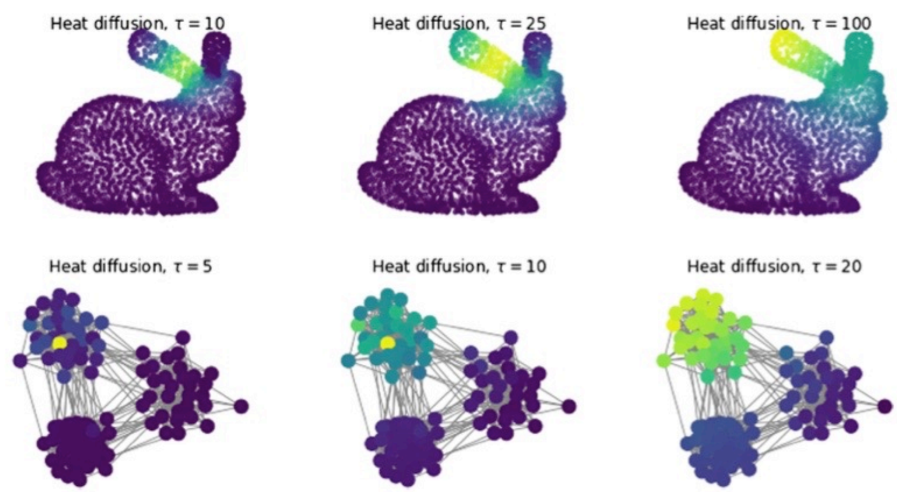


Fig. 1. Illustration of the heat diffusion over a 2-d manifold (top), and over a graph with communities (bottom), at different time τ. In both graphs, the heat spreads from node to node, following the edges. Top: the initial hot spot is a node located on the ear of the bunny. The Bunny graph is a discretization of a 2-d surface, with nodes connected to their nearest neighbours in 3 d. Bottom: The diffusion starts inside a community and quickly spreads within it.

[Ricaud et al., 2019]

Filtering of graph data

Definition

Filtering

Definition of graph filtering

We define a linear filter $\mathcal{H}$ by its function h in the Fourier domain.
It is discrete and defined on the eigenvalues $\lambda_i \to h(\lambda_i)$.

$$\widehat{\mathcal{H}(x)} = \begin{pmatrix} h(\lambda_0)\,\hat{x}(0) \\ h(\lambda_1)\,\hat{x}(1) \\ h(\lambda_2)\,\hat{x}(2) \\ \cdots \\ h(\lambda_{N-1})\,\hat{x}(N-1) \end{pmatrix} = \hat{\boldsymbol{H}}\,\hat{x} \text{ with } \hat{\boldsymbol{H}} = \begin{pmatrix} h(\lambda_0) & 0 & 0 & \dots & 0 \\ 0 & h(\lambda_1) & 0 & \dots & 0 \\ 0 & 0 & h(\lambda_2) & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & h(\lambda_{N-1}) \end{pmatrix}$$

In the node-space, the filtered signal $\mathcal{H}(x)$ can be written:

$$\mathcal{H}(x) = \chi\,\hat{\boldsymbol{H}}\,\chi^\top x$$

In term of calculus of operator on a graph, this reads

$$\mathcal{H}(x) = h(L) \cdot x$$

Example [Tremblay, Gonçalves, PB, 2017]

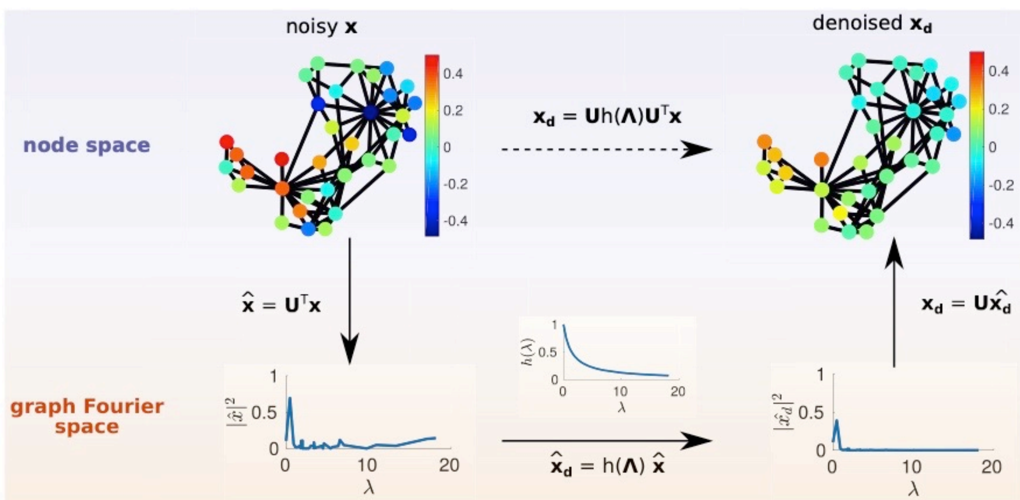


Figure 3: ***Illustration of graph filters: a denoising toy experiment.*** *The input signal* $\mathbf{x}$ *is a noisy version (additive Gaussian noise) of the low-frequency graph signal displayed in Fig. 1. We show here the filtering operation in the graph Fourier domain associated to* $\mathbf{R} = \mathbf{L_n}$.

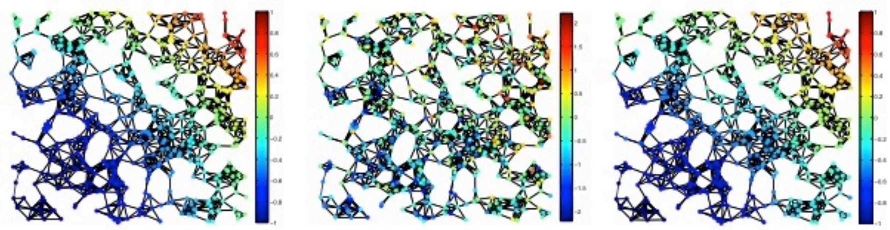
Recovery from noisy data (as an inverse problem)

Recovery of signals on graphs

[P. Vandergheynst, EPFL, 2013]

- Denoising of a signal with Tikhonov regularization

$$\arg\min_f \frac{\tau}{2}||f-y||_2^2 + f^\top L f$$



Original Noisy Denoised

Writing Tikhonov denoising as a Graph filter

- It is easy to solve this regularization problem in the spectral domain

$$\arg\min_f \frac{\tau}{2}||f-y||_2^2 + f^\top L f \Rightarrow Lf_* + \frac{\tau}{2}(f_* - y) = 0$$

- Move to the spectral domain of the Laplacian

$$\widehat{Lf_*}(i) + \frac{\tau}{2}(\hat{f}_*(i) - \hat{y}(i)) = 0, \quad \forall i \in \{0, 1, \ldots N-1\}$$

- Solution:

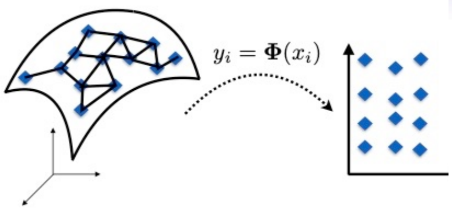
$$\hat{f}_*(i) = \frac{\tau}{\tau + 2\lambda_i}\hat{y}(i)$$

- This is a 1st-order “low pass” filtering (if the λ_i’s are considered as frequencies; here, as ω^2)

Graph embedding with harmonic analysis

- Objective of embedding: embed vertices in low dimensional space, so as to discover geometry

$$x_i \in \mathbb{R}^d \to y_i \in \mathbb{R}^k \text{ with } k < d$$



Graph embedding, Laplacian maps

- A good embedding preserves locality in the embedding space, so that nearby points are mapped nearby. It preserves smoothness.
- For that, minimize the variations of the embedding:

$$\sum_{i,j} A_{ij}(y_i - y_j)^2$$

- Laplacian eigenmaps:

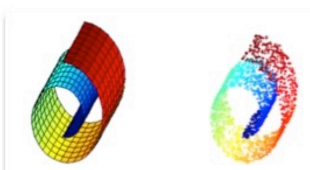
$$\begin{aligned} &\operatorname{argmin} \ \mathbf{y}^\top L\, \mathbf{y} \\ &\quad \text{such that } \mathbf{y}^\top A\, \mathbf{y} = 1 \\ &\quad\quad \text{and } \ \mathbf{y}^\top L\, \mathbf{1} = 0 \end{aligned}$$

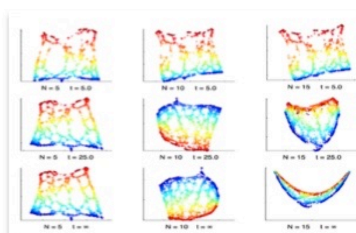
Alternative formulation:

$$L\mathbf{y} = \lambda A \mathbf{y}$$

(generalized eigenproblem)

- Some examples





[Belkin, Niyogi, 2003]