Graph Signal Processing on directed graph for modelling and learning

Pierre Borgnat

DR CNRS – Laboratoire de Physique, UMR CNRS 5672, ENS de Lyon, France
Equipe Sisyphe (Signaux, Systèmes et Physique)
et IXXI (Institut Rhônalpin des Systèmes Complexes)

Montpellier, 05/2019

Scope of the work

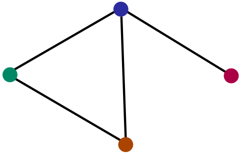
- Extension of **Graph Signal Processing** to **directed graphs**
- The motivation: tasks of signal modelling and/or learning on digraphs
- Joint work with Harry Sevi (PhD defended in Novembre 2018) and Gabriel Rilling (CEA List)
- Other collaborators (thanked also for some of the figures of the talk) :

Nicolas Tremblay, Sophie Achard, Paulo Gonçalves, Cédric Richard, Fei Hua

- Work supported by:
  - ANR-14-CE27-0001 GRAPHSIP grant
  - ACADEMICS grant in the Scientific Breakthrough Program IDEXLYON, Programme Investissements d?Avenir ANR-16-IDEX-0005

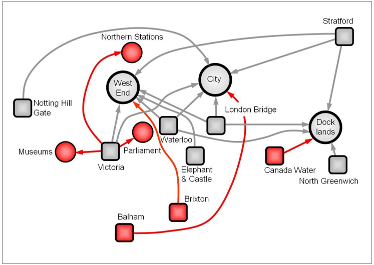
Data as graphs

- A graph $G = (V, E)$, set of nodes in V and edges in E



$V = \{blue, green, orange, red\}$ and $E = \{(b,g),(g,o),(o,b),(b,r)\}$

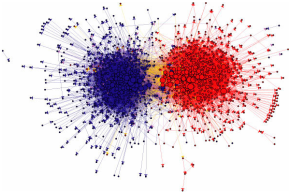
- Good to represent relations ($\in E$) between entities ($\in V$)



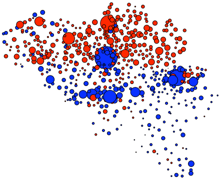
[Roth et al., 2011]

Data as graphs

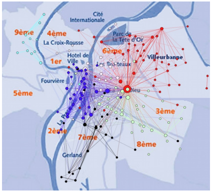
- Good to detect groups in the data ($\simeq$ clustering)



Blogosphere US 2004
[Adamic et al. 2005]

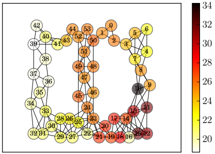


Mobile phones
[Blondel et al., 2008]

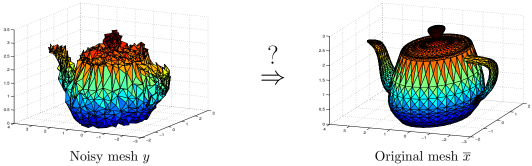


BSS Vélo'v in Lyon
[Borgnat et al., 2013]

- Good to code irregular shapes, for compression, denoising,...



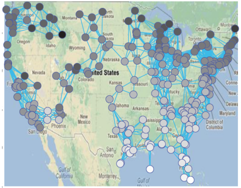
[R. Hamon et al., 2016]

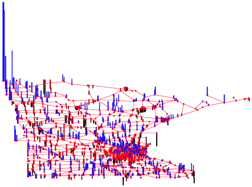


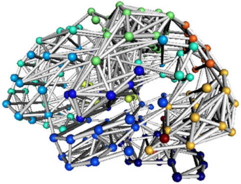
[Cours, N. Pustelnik & P.B., ENSL]

Data as graphs and graph signals

- Given a graph G, let's consider a **graph-signal** x on the nodes V.
  If $N = |V|$, we have $x \in \mathbb{R}^N$ (could be in $\mathbb{C}^N$ or multivariate)


USA Temperature


Minnesota Roads


fMRI Brain Network

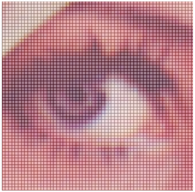


Image Grid


Color Point Cloud

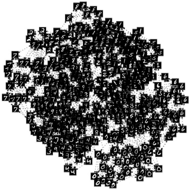
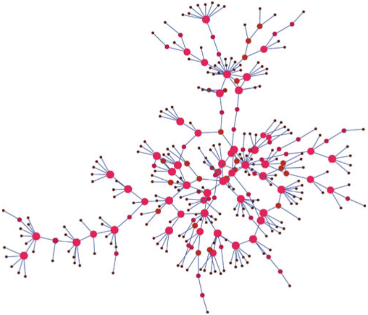


Image Database

Data as graphs and graph signals

- The question: How to apply signal processing on this data / signal ?

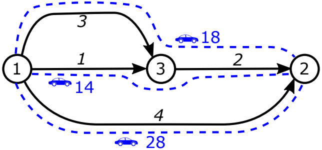
Epidemiological network



Undirected graph

[G.Ghoshal (2009), Potterat et al. (2002)]

Road network



Directed graph

[G. Michau, PB et al., 2017]

Graph Signal Processing

- “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains”
  David I Shuman ; Sunil K. Narang ; Pascal Frossard ; Antonio Ortega ; Pierre Vandergheynst,
  IEEE Signal Processing Mag., May 2013
- “Discrete Signal Processing on Graphs”
  Aliaksei Sandryhaila, Jose M. F. Moura
  IEEE Transactions on Signal Processing, April 2013
- "Graph signal processing: Overview, challenges, and applications",
  A. Ortega, P. Frossard, J. Kovacević, J.M.F. Moura, P. Vandergheynst
  Proceedings of the IEEE, 106 (5), 808-828, 2018
- “Cooperative and Graph Signal Processing”
  Ed. Petar Djuric and Cédric Richard
  Academic Press, 2018

How to define signal/data processing for graph signals?

Some basics in signal/image processing:

- Alternate representation domains of signals are useful:
  Fourier transform, DCT, time-frequency, wavelets, chirplets,...
- Among them, the Fourier transform is paramount
  Given a times series $x_n,\ n = 1, 2, ..., N$, let its Discrete Fourier Transform (DFT) be

$$\forall k \in \mathbb{Z} \quad \hat{x}_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}$$

  (for spectral analysis, filtering, computation, sampling...)

Some basics in graph-signal processing:

- Fourier transform based on the spectral domain of graph
- Develop the analog of classical SP operations by analogy:
  denoising, compression, estimation, detection, sampling,...

Two useful matrices describing graphs

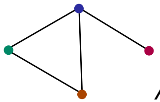
Adjacency matrix

A	adjacency matrix	$A_{ij} = 1$ if $(i,j) \in E$, else 0
d	vector of degrees	$d_i = \sum_{j \in V} A_{ij}$
D	matrix of degrees	$D = diag(d)$

Laplacian matrix (for undirected and connected G)

L or $\mathscr{L}$	laplacian matrix	$L = D - A$ or $\mathscr{L} = I - D^{-1/2} A D^{-1/2}$
(λ_i)	L's eigenvalues	$0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{N-1}$
(χ_i)	L's eigenvectors	$L\chi_i = \lambda_i \chi_i$

Also for L, $\chi_0 = \mathbf{1}$; χ_1 (Fiedler vector) is good to bisect a graph



$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} ; L = D - A = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

A Fundamental analogy for undirected graphs

[Shuman et al., *IEEE SP Mag*, 2013]

A fundamental analogy

On *any* graph, the **eigenvectors** χ_i **of the Laplacian matrix L will be considered as the Fourier modes**, and its eigenvalues λ_i the associated (squared) frequencies.

Hence, a Graph Fourier Transform is defined as:

$$\boxed{\hat{x} = \chi^\top x}$$

where $\chi = (\chi_0|\chi_1|\ldots|\chi_{N-1})$.

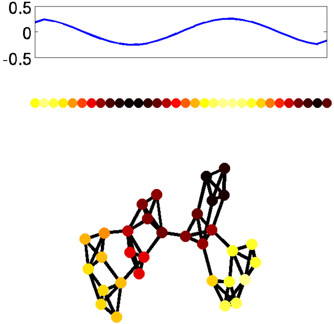
- Two ingredients:
  - **Fourier modes** = Eigenvectors χ_i (with increasing oscillations)
  - **Frequencies** = The measures of variations of an eigenvector is linked to its eigenvalue:

$$\frac{||\nabla \chi_i||^2}{||\chi_i||^2} = \lambda_i$$

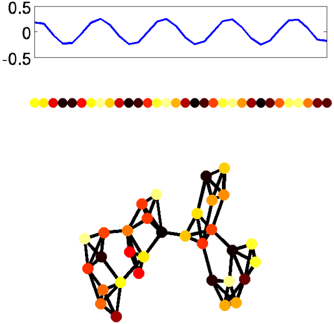
because: $\forall \mathbf{x} \in \mathbb{R}^N \quad \sum_{e=(i,j)\in E} A_{ij}(\mathbf{x}_i - \mathbf{x}_j)^2 = \mathbf{x}^\top \mathbf{L}\mathbf{x}$ is the Dirichlet norm

Fourier modes: examples in 1D and in graphs

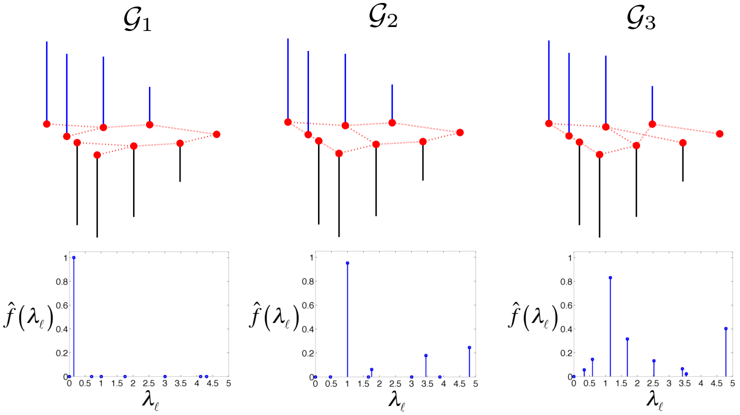
LOW FREQUENCY:



HIGH FREQUENCY:



Interplay structure - signal for assessing smoothness



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

[D. Shuman et al., 2013]

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 $\lambda_i \to h(\lambda_i)$.

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

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

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

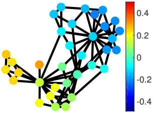
In term of calculus of operator on a graph, this reads

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

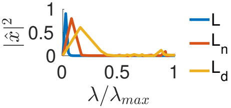
Filtering – Illustration

- Input graph signal: a noisy version of a signal, with additive Gaussian

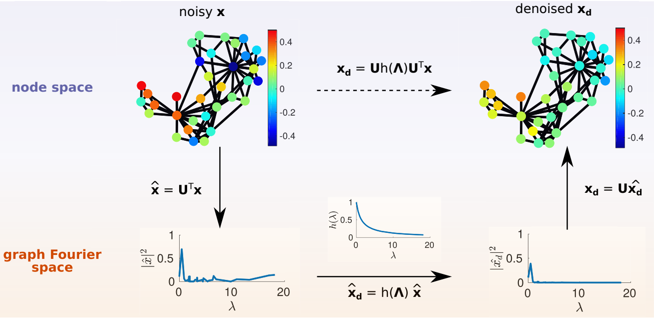
noise $\mathbf{x}$



$\hat{\mathbf{x}}$



- Denoising by filtering



[N. Tremblay, P. Gonçalves, P.B., 2018]

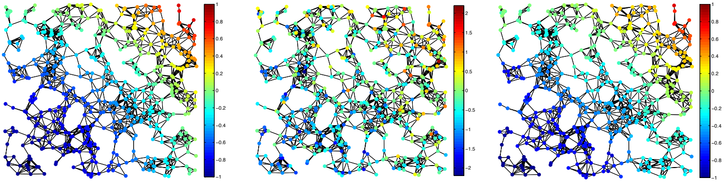
Filtering – Example for Recovery

- Denoising of a graph signal, when observing $y = x_0 + \epsilon$, formulated as an inverse problem:

$$x_* = \arg\min_x ||x - y||_2^2 + \gamma x^\top \mathbf{L} x$$

because remember that : $x^\top \mathbf{L} x = \sum_{e=(i,j)\in E} A_{ij}(x_i - x_j)^2$

- Graph-Fourier coefficients: $\hat{x} = \boldsymbol{\chi}^\top x$
- Solution: $\hat{x_*}(i) = \frac{1}{1+\gamma\lambda_i}\hat{y}(i)$ (a "1st-order low pass" filter)



Original Noisy Denoised

[P. Vandergheynst, EPFL, 2013]

Alternative versions of graph signal processing and filters

- Alternative definition of GSP:
  - Any Reference (or Shift) operator $\mathbf{R}$ can be used instead of $\mathbf{L}$
  - Discrete Signal Processing on Graphs: $\mathbf{R} = \mathbf{A}$

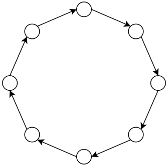
- Alternative definition of graph filters:
  - An operator $\mathbf{H}$ that commutes with the Reference operator ($\mathbf{HR} = \mathbf{RH}$) can be called a filter
  - Intuition: they share the same spectral eigenspace, hence the filter will act independently at each frequency
  - Parametric formulation: filters can be written as:

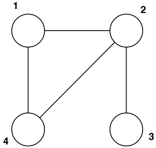
$$h(\mathbf{R}) = \sum_{k=0}^{K} h_k \mathbf{R}^k$$

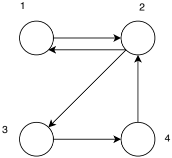
(leads to ARMA filters; to distributed implementations;...)

What about directed graphs ?

Thesis of Harry Sevi, 2018; joint work G. Rilling (CEA LIST)







Graph	cyclic	undirected	directed
Fourier Modes	$e^{i\omega t}$	χ	?
Operator		$\mathbf{L}$	?
Frequency	ω	λ	?
Variation		$\langle \chi, \mathbf{L}\chi \rangle$	?

Measure of Variations

Undirected:

$$\mathcal{D}(\boldsymbol{f}) = \frac{1}{2}\sum_{i,j} a_{ij}|f_i - f_j|^2$$

$$= \langle \boldsymbol{f}, \mathbf{L}\boldsymbol{f} \rangle$$

with

$$\mathbf{L} = \mathbf{D} - \mathbf{A}.$$

Directed:

$$\mathcal{D}^2_{\pi,\mathbf{P}}(\boldsymbol{f}) = \frac{1}{2}\sum_{i,j} \pi_i p_{ij}|f_i - f_j|^2.$$

$$= \langle \boldsymbol{f}, \mathbf{L}_{dir}\boldsymbol{f} \rangle.$$

with

$$\mathbf{L}_{dir} = \boldsymbol{\Pi} - \frac{\boldsymbol{\Pi}\mathbf{P} + \mathbf{P}^\top\boldsymbol{\Pi}}{2}$$

[Chung, 2005]

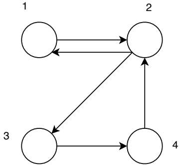
- Directed case
  - use of $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$ the random walk operator
  - and its associated stationary distribution $\boldsymbol{\pi}$,
    with the diagonal matrix $\boldsymbol{\Pi}$ associated to it
- Undirected case : $\boldsymbol{\Pi} \propto \mathbf{D} \Rightarrow \mathbf{L}_{dir} \propto \mathbf{L}$.

Fourier modes on directed graphs

Reference operator: the Random walk operator

- Random walk X_n : position X at time n.
- $\mathbf{P}_{ij} = \mathbb{P}(X_n = j | X_{n-1} = i)$ is its transition probability

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \mathbf{D}^{-1}\mathbf{A}$$



Proposed Fourier Modes:

- Eigenvectors $\mathbf{P}\boldsymbol{\xi}_k = \theta_k \boldsymbol{\xi}_k$ $\qquad \boldsymbol{\Xi} = [\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_N]$ the basis
- Fourier representation of $\boldsymbol{s}$

$$\boldsymbol{s} = \sum_k \hat{s}_k \boldsymbol{\xi}_k = \boldsymbol{\Xi}\hat{\boldsymbol{s}}.$$

where $\hat{\boldsymbol{s}} = [\hat{s}_1, \ldots, \hat{s}_N]^\top$ are the Fourier coefficients

- *Digraph Fourier Transform* :

$$\hat{\boldsymbol{s}} = \boldsymbol{\Xi}^{-1}\boldsymbol{s}.$$

- Beware : complex eigenvalues : $\theta = \alpha + \boldsymbol{i}\beta$, $|\theta| \leq 1$.

Frequency analysis of modes of **P**

Fourier Modes:

$$[\xi_1, \ldots, \xi_N]$$

Variations:

$$\mathcal{D}^2_{\pi,\mathbf{P}}(\boldsymbol{f}) = \langle \boldsymbol{f}, \mathbf{L}_{dir} \boldsymbol{f} \rangle$$

Frequency analysis:

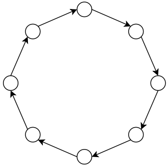
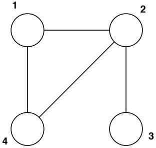
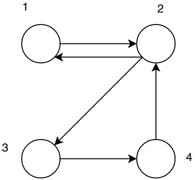
$$\frac{\mathcal{D}^2_{\pi,\mathbf{P}}(\xi)}{\langle \xi, \boldsymbol{\Pi} \xi \rangle} = 1 - \mathfrak{Re}(\theta)$$

- Let's define the **frequency** of ξ from its complex eigenvalue θ :

$$\omega = 1 - \mathfrak{Re}(\theta), \quad \omega \in [0, 2]$$

["Analyse fréquentielle et filtrage sur graphes dirigés", Sevi et al., GRETSI, 2017]

Summary of the proposed framework for directed graphs

Graphe	cyclic	undirected	directed
			
Fourier Mode	$e^{i\omega t}$	χ	ξ
Operator		$\mathbf{L}$	$\mathbf{P}$
Frequency	ω	λ	$\omega = 1 - \Re(\theta)$
Variation		$\langle \chi, \mathbf{L}\chi \rangle$	$\langle \xi, \mathbf{L}_{dir}\xi \rangle$

Comparison to other GFT for directed graphs

[**Sandryhaila et al., 2014**] DSP for graphs :

- (+) A straightforward generalisation of usual DSP
- (−) Ad-hoc definition of frequency, based on $TV^1(x) = ||x - \mathbf{A}_{norm}x||_1$
- (−) Does not generalize GSP as used on undirected graphs

[**Sardellitti et al., 2017**, **Shafipour et al., 2017**]:

Orthonormal Fourier basis related to original measures of variations

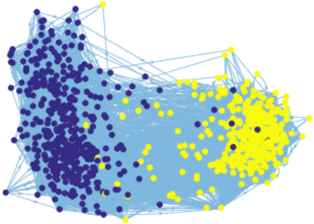
- (+) The measures of variations are interesting
- (−) The basis is found by some non convex optimisation problem
- (−) The Fourier modes are not eigenmodes of some operator
- (−) Do not generalize classical GSP on undirected graphs

Some learning tasks with GFT on digraphs

Two case studies:

- Semi-supervised learning:
  Estimation of missing data (signals) as done in [Zhou et al., ICML 2005]
- Parametric modelling of signals:
  e.g., for compression as done in [Sandryhaila et al., 2014]

Applications on the political blog data of US 2004 [Adamic et al., 2004]

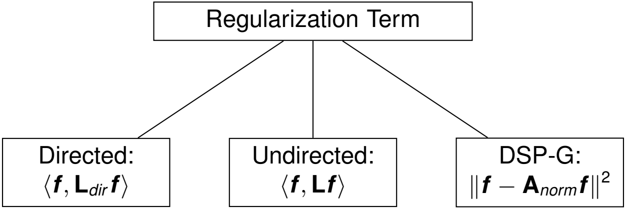


- **Node** : A political blog.
- **Edge** : A hyper-link from on blog to another (directed)
- **Signal** : The political side (democrat or republican)

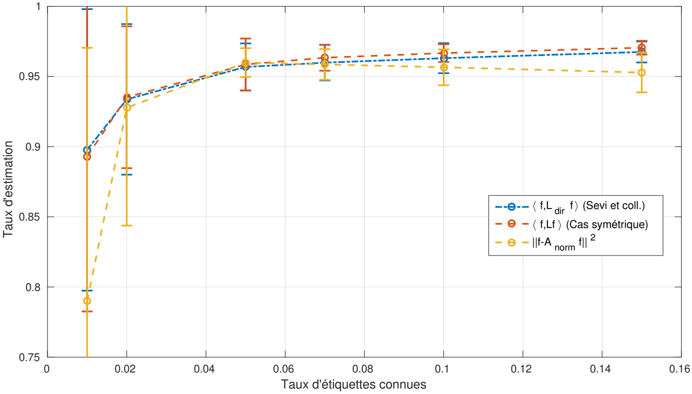
Semi-supervised learning and GSP

- Setting: some known nodes with labels ("-1", "+1"), and others with unknown labels ("0")
- The semi-supervised learning problem is to estimate labels for the unknown ones
- Solution formulated as a minimisation problem

$$f^* = \operatorname{sign}\left(\underset{f \in \ell^2(V)}{\mathbf{argmin}} \left\{ Term_{reg}(f) + Term_{data}(f) \right\} \right)$$



Example of SSL on the blog's data



Parametric modelling

Problem formulation

- Model a graph signal $\boldsymbol{f}$, e.g., for compression or inpainting
- Assumption: a partial observation $\boldsymbol{y}$ of $\boldsymbol{f}$

Objective

- Estimate a parametric modeling of $\boldsymbol{f}$
- Recover the missing data points from $\boldsymbol{y}$

Solution of the problem

- We observe $\boldsymbol{y} = \varepsilon \boldsymbol{f}$, where the $\varepsilon_k = 1$ if known, else 0
- Decide upon a **reference operator**, noted $\mathbf{R}$, first $\mathbf{R} = \mathbf{P}$ or $\mathbf{A}$
- Model the signal thanks to a parametric graph filter $\mathbf{H}$:

$$\mathbf{H}(\boldsymbol{\theta}) = \sum_{k=0}^{K} \theta_k \mathbf{R}^k, \quad \theta_k \in \mathbb{R}. \tag{1}$$

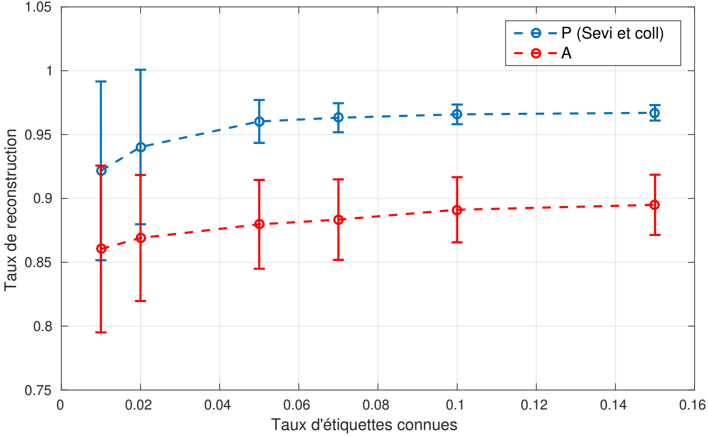
- Parameter estimation

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}=\{\theta_k\}_{k=0}^{K} \in \mathbb{R}^{K+1}}{\operatorname{argmin}} \mathbb{E}\left[\|\boldsymbol{f} - \sum_{k=0}^{K} \theta_k \mathbf{R}^k \boldsymbol{y}\|_\mu^2 \right], \tag{2}$$

- (that has well-known solution)
- **Signal model** :

$$\hat{\boldsymbol{f}}(\theta) = \sum_{k=0}^{K} \hat{\theta}_k \mathbf{R}^k \boldsymbol{y}$$

Experimental results (1)



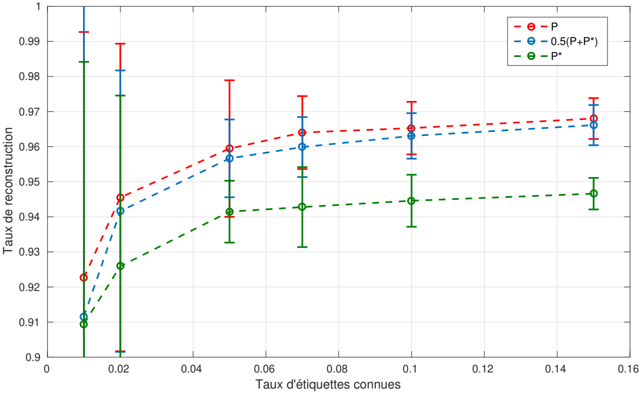
Alternative Reference Operator (1)

Other Reference operators R could be used :

- $\mathbf{P}^*$, associated to the time reversed random walk: $\mathbf{P}^* = \mathbf{\Pi}^{-1}\mathbf{P}^\top\mathbf{\Pi}$.
- $\bar{\mathbf{P}}$, the additive reversibilization of $\mathbf{P}$: $\bar{\mathbf{P}} = \dfrac{\mathbf{P} + \mathbf{P}^*}{2}$.

Prop.: $\mathbf{P}, \mathbf{P}^*, \bar{\mathbf{P}}$ lead all to DiGFT with frequency related to Variations

Experimental results (2)



Alternative Reference Operator (2)

- $\mathbf{P}^*$, associated to the time reversed random walk: $\mathbf{P}^* = \mathbf{\Pi}^{-1}\mathbf{P}^\top\mathbf{\Pi}$.
- $\bar{\mathbf{P}}$, the additive reversibilization of $\mathbf{P}$: $\bar{\mathbf{P}} = \dfrac{\mathbf{P} + \mathbf{P}^*}{2}$.

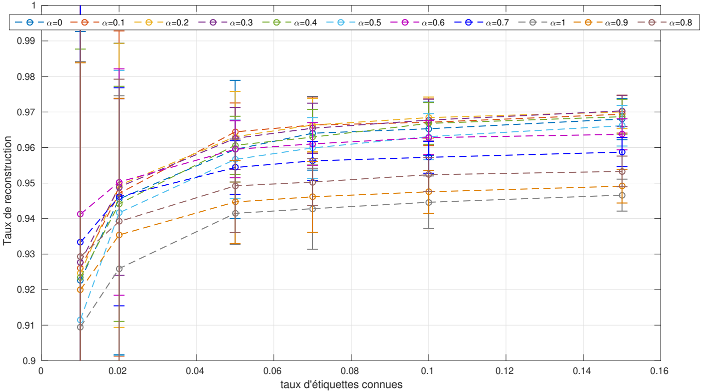
Generalization: convex combination between $\mathbf{P}$ and $\mathbf{P}^*$

$$\mathbf{P}_\alpha = (1-\alpha)\mathbf{P} + \alpha\mathbf{P}^*$$

for $\alpha \in [0,\ 1]$.

Prop.: $\mathbf{P}_\alpha$ leads all to DiGFT with frequency related to Variations

Experimental results (3)



Designing Convex Combination of Graph Filters

joint work with F. Hua, J. Chen, H. Wang, P. Gonçalves, C. Richard

- The convex combination of operators does not lead to convex optimization problem:
  - Let us consider $\mathbf{S} = \alpha \mathbf{S}_1 + (1-\alpha)\mathbf{S}_2$
  - A filter $\mathbf{H} = \sum_{\ell=0}^{L-1} h_\ell \mathbf{S}^\ell$
  - Observations of input $\mathbf{x}$ & output $\mathbf{y}$ of this filter:

$$\mathbf{y} = \mathbf{H}\mathbf{x}$$

  - The estimation of $\mathbf{h}$ and α by minimising the cost:

$$J(\mathbf{h}, \alpha) = \left\| \mathbf{y} - \sum_{\ell=0}^{L-1} h_\ell (\alpha \mathbf{S}_1 + (1-\alpha)\mathbf{S}_2)^\ell \mathbf{x} \right\|^2$$

    is non convex w.r.t. $\mathbf{h}$ and α!
- This would still not be the case by adding two filters:
  -

$$J'(\mathbf{h}, \alpha) = \left\| \mathbf{y} - [\alpha(\sum_{\ell=0}^{L-1} h_{1,\ell} \mathbf{S}_1^\ell) + (1-\alpha) \sum_{\ell=0}^{L-1} h_{2,\ell} \mathbf{S}_2^\ell] \mathbf{x} \right\|^2$$

Designing Convex Combination of Graph Filters

joint work with F. Hua, J. Chen, H. Wang, P. Gonçalves, C. Richard

- Solution: use the approach of ["Simple MKL", Rakotomamonjy et al., JLMR 2008]
- The combination is now of different filters:

$$\mathbf{H} = \sum_{\ell=0}^{L-1} h_{1,\ell}\mathbf{S}_1^\ell + h_{2,\ell}\mathbf{S}_2^\ell$$

- The combination is now regularised and controlled by minimizing a balance between the norms of the vectors of filter coefficients:

$$\boldsymbol{h}_1^*, \boldsymbol{h}_2^*, \alpha^* = \operatorname*{arg\,min}_{\boldsymbol{h}_1,\boldsymbol{h}_2,\alpha} \quad \frac{1}{2}\left(\frac{\|\boldsymbol{h}_1\|^2}{\alpha} + \frac{\|\boldsymbol{h}_2\|^2}{1-\alpha}\right) + \frac{1}{2\mu}\sum_{k=1}^{N} e_k^2$$

$$\text{subject to :}$$

$$e_k = y_k - \boldsymbol{h}_1^\top \boldsymbol{m}_{1,k} - \boldsymbol{h}_2^\top \boldsymbol{m}_{2,k}, \quad k \in \{1,\ldots,N\}$$

$$0 < \alpha < 1,$$

where $m_{1,k}^\top$ is the k-th row of the matrix $\mathbf{M}_1$, itself is obtained by concatenation as $[\mathbf{M}_1]_{\cdot,\ell} = \mathbf{S}_1^{\ell-1}\mathbf{x}$; idem for $m_{2,k}^\top$

Designing Convex Combination of Graph Filters

joint work with F. Hua, J. Chen, H. Wang, P. Gonçalves, C. Richard

- The problem is now an optimization problem that is jointly convex w.r.t. $\mathbf{h}_1$, $\mathbf{h}_2$ and α
- Solution obtained with a two-step procedure, w.r.t. $\mathbf{h}_1$, $\mathbf{h}_2$ and then α

Algorithm 1

Input: $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{S}_1, \boldsymbol{S}_2, L$.
Initialize: randomly choose $0 < \alpha^*_{-1} < 1$, compute $\boldsymbol{M}_1, \boldsymbol{M}_2$.
Repeat:
1: solve (22) with a generic QP solver to get $\boldsymbol{\lambda}^*$
compute $\boldsymbol{h}_1^*, \boldsymbol{h}_2^*$ from (21),
2: update α_i^* by using (25).
Until: stopping condition is satisfied.
Output: $\boldsymbol{h}_1^*, \boldsymbol{h}_2^*, \alpha^*$.

$$\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}\in\mathbb{R}^N} -\frac{1}{2}\boldsymbol{\lambda}^\top(\boldsymbol{R}_\alpha + \mu\boldsymbol{I})\boldsymbol{\lambda} + \boldsymbol{\lambda}^\top\boldsymbol{y} \quad \text{with } \boldsymbol{R}_\alpha = \alpha\boldsymbol{M}_1\boldsymbol{M}_1^\top + (1-\alpha)\boldsymbol{M}_2\boldsymbol{M}_2^\top. \tag{22}$$

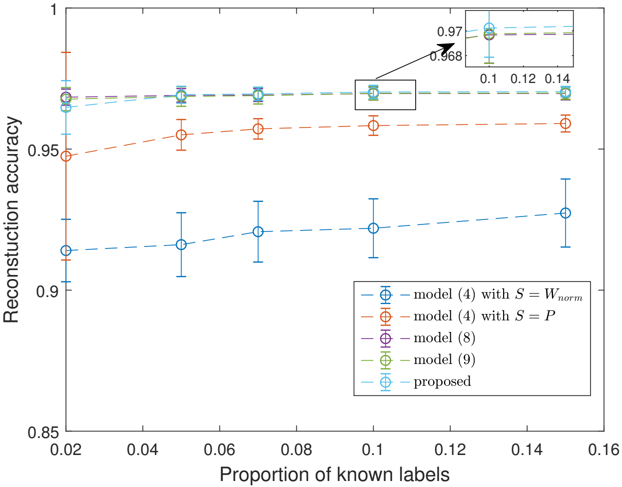
$$\begin{cases} \boldsymbol{h}_1^* &= \alpha\sum_{k=1}^N \lambda_k^*\boldsymbol{m}_{1,k} \\ \boldsymbol{h}_2^* &= (1-\alpha)\sum_{k=1}^N \lambda_k^*\boldsymbol{m}_{2,k}\,. \\ e_k^* &= \mu\lambda_k^* \end{cases} \tag{21}$$

$$\alpha_i^* = \left(1 + \frac{1-\alpha_{i-1}^*}{\alpha_{i-1}^*}\sqrt{\frac{\boldsymbol{\lambda}^{*\top}\boldsymbol{M}_2\boldsymbol{M}_2^\top\boldsymbol{\lambda}^*}{\boldsymbol{\lambda}^{*\top}\boldsymbol{M}_1\boldsymbol{M}_1^\top\boldsymbol{\lambda}^*}}\right)^{-1} \tag{25}$$

Designing Convex Combination of Graph Filters

Application: Signal recovery on the political blogs

- Combined operators: $\mathbf{S}_1 = \mathbf{P}$ and $\mathbf{S}_1 = \mathbf{P}^*$
- Accuracy result:



Perspectives for GSP on directed graphs

- A full framework to generalize Laplacian-based approaches to digraphs,

  - using random walk (or generalisations $\mathbf{P}_\alpha$) as Reference operator
  - and $\mathbf{L}_{dir}$ to measure variations and define frequency

- Re-interpretation of SSL ; Improvement of parametric modelling

- More developments:

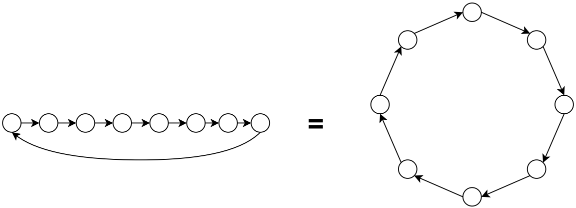
  Spectral wavelets and diffusion wavelets with **P** on digraphs
  see [H. Sevi, G. Rilling, P.B., arXiv:1811.11636]

- Recent interests in combining that and more machine learning
  cf. ACADEMICS project (SB IDEXLYON)

- Contact and more information:

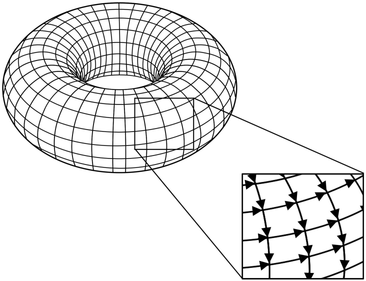
  http://perso.ens-lyon.fr/pierre.borgnat

On the directed cyclic graph

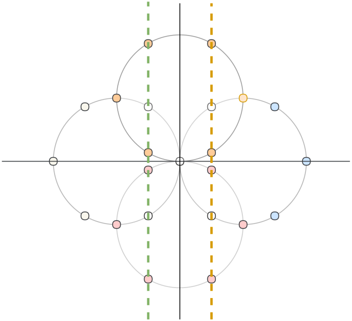


	Classic DSP		Directed cycle graph
Eigenvectors	$e^{i\omega t}, e^{-i\omega t}$	=	$\theta^t, \bar{\theta}^t$
Eigenvalues	$e^{i\omega}, e^{-i\omega}$	=	$\theta, \bar{\theta}$
Frequencies	$\omega, -\omega$	$\neq$	$\theta, \bar{\theta} = (1-\omega) \pm i\beta$

On a directed torus graph



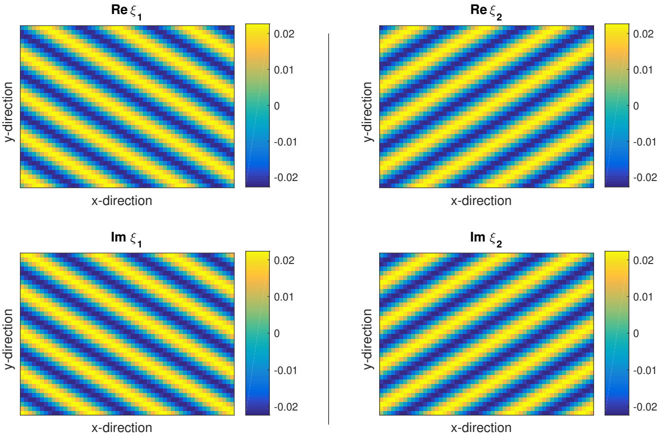
Directed torus graph



Eigenvalues of **P**.

On a directed torus graph

We show 2 eigenmodes of same frequency and different (non conjugate) imaginary parts



Further numerical explorations of graph signal modelling

- Results depends on the sampling law for ε (where $\boldsymbol{t} = \varepsilon \boldsymbol{f}$)
- A limit of choosing **P**: it requires a strongly connected graph...
- 1) use connected components,
- or 2) modify the graph
  - add a small rank-one perturbation (Cons: non-sparse)
  - construct the "google" matrix:
    complete dangling nodes (i.e., nodes with $d^{out} = 0$)
    and then add a probability of jumping anywhere

Experimental results (4)