HAL
open science

Smoothing graph signals via random spanning forests

Yusuf Yigit Pilavci, Pierre-Olivier Amblard, Simon Barthelme, Nicolas Tremblay

To cite this version:

Yusuf Yigit Pilavci, Pierre-Olivier Amblard, Simon Barthelme, Nicolas Tremblay. Smoothing graph signals via random spanning forests. ICASSP 2020 - IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, May 2020, Barcelone (virtual), Spain. 10.1109/ICASSP40776.2020.9054497 . hal-02319175v2

HAL Id: hal-02319175

https://hal.science/hal-02319175v2

Submitted on 5 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SMOOTHING GRAPH SIGNALS VIA RANDOM SPANNING FORESTS

Yusuf Y. Pilavci, Pierre-Olivier Amblard, Simon Barthelmé, Nicolas Tremblay

CNRS, Univ. Grenoble Alpes, Grenoble INP, GIPSA-lab, Grenoble, France

ABSTRACT

Another facet of the elegant link between random processes on graphs and Laplacian-based numerical linear algebra is uncovered: based on random spanning forests, novel Monte-Carlo estimators for graph signal smoothing are proposed. These random forests are sampled efficiently via a variant of Wilson's algorithm –in time linear in the number of edges. The theoretical variance of the proposed estimators are analyzed, and their application to several problems are considered, such as Tikhonov denoising of graph signals or semi-supervised learning for node classification on graphs.

***Index Terms*—** graph signal processing, smoothing, random spanning forests

1. INTRODUCTION

Tikhonov denoising of graph signals [1, 2], semi-supervised learning for node classification in graphs [3], post-sampling graph signal reconstruction [4] are a few examples falling in the following class of optimization problems. Given a graph signal $\mathbf{y} = (y_1|\dots|y_n)^t \in \mathbb{R}^n$ where y_i is the value measured at node i of a graph, one considers the problem

$$\hat{\mathbf{x}} = \underset{\mathbf{z}\in\mathbb{R}^n}{\operatorname{argmin}}\, q\,||\mathbf{y}-\mathbf{z}||^2 + \mathbf{z}^t\mathsf{L}\mathbf{z}, \tag{1}$$

where L is the Laplacian of the graph and $q > 0$ a parameter tuning the trade-off between a data-fidelity term $||\mathbf{y}-\mathbf{z}||^2$ –encouraging the solution $\hat{\mathbf{x}}$ to be close to $\mathbf{y}$– and a smoothness term $\mathbf{z}^t\mathsf{L}\mathbf{z}$ –encouraging the solution $\hat{\mathbf{x}}$ to vary slowly along any path of the graph. In Tikhonov denoising, $\mathbf{y}$ is a noisy measurement of an underlying signal $\mathbf{x}$ that one wants to recover. In semi-supervised learning (SSL), $\mathbf{y}$ are known labels that one wants to propagate on the graph to classify all the nodes in different classes (see Section 4 for more details).

This optimization problem admits an explicit solution:

$$\hat{\mathbf{x}} = \mathsf{K}\mathbf{y} \quad \text{with} \quad \mathsf{K} = q(q\mathsf{I}+\mathsf{L})^{-1} \in \mathbb{R}^{n\times n}, \tag{2}$$

that requires the inversion of a regularized Laplacian $q\mathsf{I}+\mathsf{L}$, where I is the identity matrix. Computing K costs $\mathcal{O}(n^3)$ elementary operations and becomes prohibitive as n increases.

This work was partly funded by the ANR GenGP (ANR-16-CE23-0008), the ANR GraVa (ANR-18-CE40-0005), the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01), the CNRS PEPS I3A (Project RW4SPEC), the Grenoble Data Institute (ANR-15-IDEX-02), the LIA CNRS/Melbourne Univ Geodesic, and the MIAI chair "LargeDATA at UGA."

State-of-the-art. For large n (say $\geq 10^4$), the state-of-the-art includes iterative methods such as conjugate gradient with preconditioning [5] where L is only accessed via matrix-vector multiplications of the form $\mathsf{L}\mathbf{z}$, and polynomial approximation methods [6] that approximate K by a low-order polynomial in L. Both classes of methods enable to compute $\hat{\mathbf{x}}$ in time linear in $|\mathcal{E}|$, the number of edges of the graph.

Contribution. We introduce two Monte-Carlo estimators of $\hat{\mathbf{x}}$ based on random spanning forests, that:

- show another facet of the elegant link between random processes on graphs and Laplacian-based numerical linear algebra, such as in [7, 8, 9]
- scale linearly in $|\mathcal{E}|$ and thus useful on very large graphs
- can be implemented in a fully distributed fashion: as long as each node is able to communicate with its neighbours, the result can be obtained without centralized knowledge of the graph's structure L (see the implementation paragraph at the end of Section 3).

The Julia code implementing these estimators and reproducing this papers' results is available on the authors' website[1].

Structure of the paper. We provide the necessary notations and background in Section 2. In Section 3, we detail the two novel estimators before generalizing them to cases where K is of the form $\mathsf{K} = (\mathsf{Q}+\mathsf{L})^{-1}\mathsf{Q}$ with Q a positive diagonal matrix. The experimental Section 4 gives an illustration on image denoising before showing how to use our estimators in the context of SSL. We conclude in Section 5.

2. BACKGROUND

Notations and preliminary definitions. We consider undirected graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathsf{W})$ where $\mathcal{V}$ is the set of $|\mathcal{V}| = n$ nodes, $\mathcal{E}$ the set of $|\mathcal{E}|$ edges, and $\mathsf{W} \in \mathbb{R}^{n\times n}$ the symmetric weighted adjacency matrix. We denote by $d_i = \sum_j \mathsf{W}_{ij}$ the degree of node i, $\mathbf{d} = (d_1, d_2, \dots, d_n)^t \in \mathbb{R}^n$ the degree vector, and $\mathsf{D} = \operatorname{diag}(\mathbf{d}) \in \mathbb{R}^{n\times n}$ the diagonal degree matrix. In this paper, we consider the Laplacian matrix defined as $\mathsf{L} = \mathsf{D}-\mathsf{W} \in \mathbb{R}^{n\times n}$. L is positive semi-definite (PSD) [7] and its eigenvalues can be ordered as $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. We also consider graph signals $\mathbf{x} \in \mathbb{R}^n$ defined over the nodes $\mathcal{V}$: $x_i = x(i)$ is the signal's value at node i. We denote by

[1] www.gipsa-lab.fr/~nicolas.tremblay/files/graph_smoothing_via_RSF.zip

$\mathbf{1} \in \mathbb{R}^n$ the constant vector equal to 1, and by $\boldsymbol{\delta}_i \in \mathbb{R}^n$ the Kronecker delta such that $\delta_i(j) = 1$ if $j = i$, and 0 otherwise.

A tree of $\mathcal{G}$ is understood as a connected subgraph of $\mathcal{G}$ that does not contain cycles. A rooted tree of $\mathcal{G}$, *i.e.*, a tree of $\mathcal{G}$ whose edges are all oriented towards one node called root, is generically denoted by τ. A rooted forest of $\mathcal{G}$, *i.e.*, a set of disjoint rooted trees, is generically denoted by ϕ. In the following, we only consider rooted trees and rooted forests, and thus simply refer to them as trees and forests. Also, $\rho(\phi)$ stands for the set of roots of the trees in ϕ. As such, $|\rho(\phi)|$ is the total number of trees in ϕ. We define the function $r_\phi : \mathcal{V} \to \mathcal{V}$ such that $r_\phi(i)$ is the root of the tree containing node i in the forest ϕ. Node i is said to be rooted in $r_\phi(i)$.

Random spanning forests (RSFs). Let us recall that a spanning tree (resp. forest) of $\mathcal{G}$ is a tree (resp. forest) of $\mathcal{G}$ that spans all nodes in $\mathcal{V}$. Now, for a given graph $\mathcal{G}$, there exists in general many spanning trees (and even more spanning forests). Probabilists and computer scientists have been studying several distributions over spanning trees and forests in the past. A classical distribution over spanning trees is:

$$\mathbb{P}(T = \tau) \propto \prod_{(ij)\in\tau} \mathsf{W}_{ij}. \tag{3}$$

Sampling from this distribution yields a so-called uniform[2] spanning tree (UST) T, and can be efficiently performed by Wilson's algorithm [10] via loop-erased random walks.

We focus here on the following parametrized distribution over spanning forests:

$$\mathbb{P}(\Phi_q = \phi) \propto q^{|\rho(\phi)|} \prod_{\tau\in\phi} \prod_{(ij)\in\tau} \mathsf{W}_{ij}, \quad q \in \mathbb{R}^{+*}. \tag{4}$$

Sampling from this distribution yields a RSF Φ_q and is efficiently performed (in time $\mathcal{O}(|\mathcal{E}|/q)$) via a variant of Wilson's algorithm [11]. Many properties of Φ_q are known [11, 12]. For instance, the probability that node i is rooted in j is (see lemma 3.3 in [12])

$$\mathbb{P}\left(r_{\Phi_q}(i) = j\right) = \mathsf{K}_{ij}. \tag{5}$$

3. RSF-BASED ESTIMATORS

Given a signal $\mathbf{y}$, our goal is to compute $\hat{\mathbf{x}} = \mathsf{K}\mathbf{y}$ without computing explicitly K, that is, without inverting $q\mathsf{I} + \mathsf{L}$. We define two Monte-Carlo estimators of $\hat{\mathbf{x}}$, both based on RSFs.

A first estimator of $\hat{\mathbf{x}}$. Let us consider a realisation Φ_q of RSF and define the estimator $\tilde{\mathbf{x}}$ as

$$\forall i \quad \tilde{x}(i) = y\left[r_{\Phi_q}(i)\right]. \tag{6}$$

Proposition 1. *$\tilde{\mathbf{x}}$ is unbiased: $\mathbb{E}[\tilde{\mathbf{x}}] = \hat{\mathbf{x}}$. Moreover:*

$$\mathbb{E}\left(||\tilde{\mathbf{x}} - \hat{\mathbf{x}}||^2\right) = \sum_i \mathrm{var}(\tilde{x}(i)) = \mathbf{y}^t(\mathsf{I} - \mathsf{K}^2)\mathbf{y}.$$

[2]this terminology comes from the fact that in unweighted graphs, this distribution reduces to the uniform distribution over all spanning trees

Proof. Let $i \in \mathcal{V}$. One has, using Eq. (5):

$$\mathbb{E}[\tilde{x}(i)] = \mathbb{E}\left[y\left[r_{\Phi_q}(i)\right]\right] = \sum_j \mathbb{P}(r_{\Phi_q}(i) = j) y_j \tag{7}$$

$$= \sum_j \mathsf{K}_{ij} y_j = \boldsymbol{\delta}_i^t \mathsf{K}\mathbf{y} = \hat{x}(i). \tag{8}$$

$\tilde{\mathbf{x}}$ is thus unbiased. A similar calculation yields $\mathbb{E}\left[\tilde{x}(i)^2\right] = \boldsymbol{\delta}_i^t \mathsf{K}\mathbf{y}^{(2)}$ where $\forall k, y_k^{(2)} = y_k^2$, such that the variance verifies:

$$\mathrm{var}(\tilde{x}(i)) = \mathbb{E}\left[\tilde{x}(i)^2\right] - \mathbb{E}\left[\tilde{x}(i)\right]^2 = \boldsymbol{\delta}_i^t \mathsf{K}\mathbf{y}^{(2)} - (\boldsymbol{\delta}_i^t \mathsf{K}\mathbf{y})^2.$$

Summing over all nodes yields:

$$\sum_i \mathrm{var}(\tilde{x}(i)) = \mathbf{1}^t \mathsf{K}\mathbf{y}^{(2)} - \mathbf{y}^t \mathsf{K}^2 \mathbf{y}.$$

Noticing that $\mathbf{1}$ is an eigenvector of K with eigenvalue 1 (as $\mathbf{1}$ is an eigenvector of L with eigenvalue 0) ends the proof. □

An improved estimator of $\hat{\mathbf{x}}$. A random forest Φ_q contains $|\rho(\Phi_q)|$ trees that we enumerate from 1 to $|\rho(\Phi_q)|$. Consider $\mathcal{V}_k \subset \mathcal{V}$ the subset of nodes that are in the k-th tree. As Φ_q is a spanning forest, $\mathcal{P} = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_{|\rho(\Phi_q)|}\}$ is a partition of $\mathcal{V}$ (*i.e.*, a set of disjoint subsets that cover $\mathcal{V}$). Let t be the tree membership function that associates to each node i the tree number $t(i)$ to which it belongs. For instance, $|\mathcal{V}_{t(i)}|$ is the size of the tree containing node i. We define a second estimator as

$$\forall i \quad \bar{x}(i) = \frac{1}{|\mathcal{V}_{t(i)}|} \sum_{j\in\mathcal{V}_{t(i)}} y_j. \tag{9}$$

Proposition 2. *$\bar{\mathbf{x}}$ is unbiased: $\mathbb{E}[\bar{\mathbf{x}}] = \hat{\mathbf{x}}$. Moreover:*

$$\mathbb{E}\left(||\bar{\mathbf{x}} - \hat{\mathbf{x}}||^2\right) = \sum_i \mathrm{var}(\bar{x}(i)) = \mathbf{y}^t(\mathsf{K} - \mathsf{K}^2)\mathbf{y}.$$

Proof. Let us denote by π the function that takes as entry a forest ϕ and outputs its associated partition. Let us also define $s_{ij} = 1$ if $t(i) = t(j)$, and 0 otherwise. We need the Proposition 2.3 of [11]. Fixing a partition $\mathcal{P}$ of $\mathcal{V}$, one has:

$$\mathbb{P}\left(r_{\Phi_q}(i) = j | \pi(\Phi_q) = \mathcal{P}\right) = \frac{s_{ij}}{|\mathcal{V}_{t(i)}|}.$$

In words, this means that given a partition, the distribution of the root within each part $\mathcal{V}_l$ is uniform over $\mathcal{V}_l$. Also, note that $\mathsf{K}_{ij} = \mathbb{P}\left(r_\phi(i) = j\right)$ can be written:

$$\begin{aligned}
\mathsf{K}_{ij} &= \sum_{\mathcal{P}} \mathbb{P}\left(r_{\Phi_q}(i) = j | \pi(\Phi_q) = \mathcal{P}\right) \mathbb{P}\left(\pi(\Phi_q) = \mathcal{P}\right) \\
&= \sum_{\mathcal{P}} \frac{s_{ij}}{|\mathcal{V}_{t(i)}|} \mathbb{P}\left(\pi(\Phi_q) = \mathcal{P}\right) \\
&= \sum_{\mathcal{P}} \frac{s_{ij}}{|\mathcal{V}_{t(i)}|} \sum_{\phi \text{ s.t. } \pi(\phi)=\mathcal{P}} \mathbb{P}\left(\Phi_q = \phi\right) \\
&= \sum_{\phi} \mathbb{P}\left(\Phi_q = \phi\right) \frac{s_{ij}}{|\mathcal{V}_{t(i)}|} = \mathbb{E}\left(\frac{s_{ij}}{|\mathcal{V}_{t(i)}|}\right).
\end{aligned} \tag{10}$$

Given a RSF Φ_q, define the matrix $\mathsf{S}_{ij} = \frac{s_{ij}}{|\mathcal{V}_{t(i)}|} \in \mathbb{R}^{n\times n}$. Eq. (10) translates as: $\mathbb{E}(\mathsf{S}) = \mathsf{K}$. Thus, $\forall i \in \mathcal{V}$:

$$\mathbb{E}\left[\bar{x}(i)\right] = \mathbb{E}\left[\frac{1}{|\mathcal{V}_{t(i)}|}\sum_{j\in\mathcal{V}_{t(i)}} y_j\right] \tag{11}$$

$$= \mathbb{E}\left[\boldsymbol{\delta}_i^t \mathsf{S}\mathbf{y}\right] = \boldsymbol{\delta}_i^t \mathbb{E}\left[\mathsf{S}\right]\mathbf{y} = \boldsymbol{\delta}_i^t \mathsf{K}\mathbf{y} = \hat{\mathbf{x}}_i. \tag{12}$$

The estimator $\bar{\mathbf{x}}$ is thus unbiased. Note that one also has:

$$\mathbb{E}\left[\bar{x}(i)^2\right] = \mathbb{E}\left[\left(\boldsymbol{\delta}_i^t \mathsf{S}\mathbf{y}\right)^2\right] = \mathbf{y}^t \mathbb{E}\left[\mathsf{S}^t \boldsymbol{\delta}_i \boldsymbol{\delta}_i^t \mathsf{S}\right]\mathbf{y}, \tag{13}$$

such that

$$\sum_i \mathrm{var}(\bar{x}(i)) = \sum_i \mathbf{y}^t \mathbb{E}\left[\mathsf{S}^t \boldsymbol{\delta}_i \boldsymbol{\delta}_i^t \mathsf{S}\right]\mathbf{y} - (\boldsymbol{\delta}_i^t \mathsf{K}\mathbf{y})^2 \tag{14}$$

$$= \mathbf{y}^t\left[\mathbb{E}\left[\mathsf{S}^t\mathsf{S}\right] - \mathsf{K}^2\right]\mathbf{y}. \tag{15}$$

Note that $\mathsf{S}^t\mathsf{S} = \mathsf{S}$, *i.e.*, $\mathbb{E}\left[\mathsf{S}^t\mathsf{S}\right] = \mathsf{K}$, finishing the proof. □

Rao-Blackwellisation. Note that $\bar{\mathbf{x}}$ is a Rao-Blackwellisation of $\tilde{\mathbf{x}}$ where the sufficient statistic is the partition induced by the RSF. As such, $\bar{\mathbf{x}}$ is necessarily an improvement over $\tilde{\mathbf{x}}$. This improvement can also be observed in the variance equations of both propositions: as K is PSD with eigenvalues between 0 and 1, one has: $\forall \mathbf{y};\ \mathbf{y}^t(\mathsf{K}-\mathsf{K}^2)\mathbf{y} \leq \mathbf{y}^t(\mathsf{I}-\mathsf{K}^2)\mathbf{y}$.

Tikhonov denoising. Let $\mathbf{y} = \mathbf{x} + \boldsymbol{\epsilon}$ be a noisy measurement of a signal $\mathbf{x}$ that one wishes to recover. Assuming that the measurement noise $\boldsymbol{\epsilon}$ is Gaussian with covariance $\mathbb{E}_{\boldsymbol{\epsilon}}\left(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^t\right) = \gamma^2\mathsf{I}$, one can write (for instance for the second estimator):

$$\mathbb{E}_{\boldsymbol{\epsilon}}\left[\mathbb{E}\left(||\bar{\mathbf{x}} - \hat{\mathbf{x}}||^2\right)\right] = \mathbf{x}^t(\mathsf{K}-\mathsf{K}^2)\mathbf{x} + \gamma^2 \mathrm{Tr}\left(\mathsf{K}-\mathsf{K}^2\right)$$
$$= ||\bar{\mathsf{F}}\mathbf{x}||_2^2 + \gamma^2\mathrm{Tr}\left(\bar{\mathsf{F}}^2\right),$$

where $\bar{\mathsf{F}} = \mathsf{U}\bar{f}(\Lambda)\mathsf{U}^t$ is a graph bandpass filter [1, 13] with frequency response $\bar{f}(\lambda) = \frac{\sqrt{q\lambda}}{q+\lambda}$. The second term depends on the noise level γ. The first term, however, depends on the original signal $\mathbf{x}$ filtered by $\bar{f}$: the Fourier components of $\mathbf{x}$ associated with graph frequencies around $\lambda = q$ (maximizing $\bar{f}$) are thus harder to denoise than the ones close to 0 or λ_n.

In practice. For a given q, N independent RSFs are sampled –in time $\mathcal{O}(\frac{N|\mathcal{E}|}{q})$. Each RSF provides an independent estimate of $\hat{\mathbf{x}}$, and all N estimates are finally averaged.

Generalisation. Instead of estimating results of the form $(q\mathsf{I}+\mathsf{L})^{-1}q\mathbf{y}$, one may need to estimate results of the form $(\mathsf{Q}+\mathsf{L})^{-1}\mathsf{Q}\mathbf{y}$ where $\mathsf{Q} = \mathrm{diag}(\mathbf{q})$ is a diagonal matrix, with $\mathbf{q} = (q_1|\ldots|q_n)^t \in (0,+\infty)^n$. In order to tackle this case, one considers the following distribution over forests:

$$\mathbb{P}(\Phi_{\mathsf{Q}} = \phi) \propto \prod_{r\in\rho(\phi)} q_r \prod_{\tau\in\phi}\prod_{(ij)\in\tau} \mathsf{W}_{ij}, \tag{16}$$

that generalizes the distribution of Eq. (4). One can efficiently sample from this distribution –also via a variant of Wilson's algorithm (see the next paragraph). The introduced estimators generalize naturally to this case. In fact, given a RSF Φ, their formulation is exactly the same, the sole difference stemming from the distribution from which Φ is drawn from. Propositions 1 and 2 are still correct (proofs are omitted due to lack of space) for $\mathsf{K} = (\mathsf{Q}+L)^{-1}\mathsf{Q}$ instead of $\mathsf{K} = (q\mathsf{I}+L)^{-1}q\mathsf{I}$.

Implementation. In a nutshell, an algorithm sampling from the distribution of Eq. (16) i/ adds a node Δ to the graph and connects it to each node i with weight q_i; ii/ runs Wilson's `RandomTreeWithRoot` algorithm (based on loop-erased random walks –see Figure 1 of [10]) on this augmented graph to sample a UST T rooted in Δ; iii/ cuts the edges connected to Δ in T yielding a RSF Φ_{Q}. Then, the estimator $\tilde{\mathbf{x}}$ associates to each node i the value of $\mathbf{y}$ at its root $r_{\Phi_{\mathsf{Q}}}(i)$, whereas the estimator $\bar{\mathbf{x}}$ associates to each node i the average of $\mathbf{y}$ over all the nodes belonging to its tree. All these operations can be done in a distributed fashion: no centralized knowledge of the graph is needed. Also, once the RSF is sampled, the computations involved for the estimators are not only distributed but can also be made in parallel (within each tree of the forest). To give an order of magnitude of computation times, for a random vector $\mathbf{y}$ and $\mathbf{q} = \mathbf{1}$, our naive Julia implementation of $\bar{\mathbf{x}}$ on a graph with $n = 10^5$ (resp. 10^6) nodes and $|\mathcal{E}| = 10^6$ (resp 10^7) edges runs in average in 35 (resp. 550) ms on a laptop. These times are to compare to the optimized built-in sparse matrix multiplication $\mathsf{L}\mathbf{y}$ running in 6 (resp. 115) ms, which is the building block of both conjugate gradient and polynomial approximation methods stated in the introduction. Our methods are thus comparable to the state-of-the-art in computation time.

4. EXPERIMENTS

Illustration on an image. Fig. 1 shows an image denoising example on a 64×64 grayscale image. Constant irregular patches are observed on the realisation of $\tilde{\mathbf{x}}(N=1)$: they are the trees of the associated RSF realisation. Also, as expected, $\bar{\mathbf{x}}$ converges faster than $\tilde{\mathbf{x}}$ (as N increases) for all values of q.

Illustration on SSL. The goal of SSL is to infer the label (or class) of all the nodes of a graph given a few pre-labelled nodes. Consider a partial labeling $\mathbf{Y} = (\mathbf{y}_1|\mathbf{y}_2|\ldots|\mathbf{y}_k) \in \mathbb{R}^{n\times k}$ of the nodes, where k is the number of classes and $\mathbf{y}_l(i)$ is equal to 1 if node i is *a priori* known to belong to class l and 0 otherwise. The objective is to find k classification functions $\{\mathbf{f}_l\}_{l=1,\ldots,k}$ such that each $\mathbf{f}_l$ is on the one hand close to the labeling function $\mathbf{y}_l$ and on the other hand smooth on the graph, with a trade-off given by a parameter $\mu > 0$. Depending on the choice of Laplacian used to define this graph smoothness (in the following, $\sigma = 1$ corresponds to the combinatorial Laplacian, $\sigma = 1/2$ to the normalized Laplacian, $\sigma = 0$ to the random walk Laplacian), the explicit formulation of $\mathbf{f}_l$ can be written as (Prop. 2.2 of [3]): $\mathbf{f}_l = \frac{\mu}{2+\mu}\left(\mathsf{I} - \frac{2}{2+\mu}\mathsf{D}^{-\sigma}\mathsf{W}\mathsf{D}^{\sigma-1}\right)^{-1}\mathbf{y}_l$. Note that this can be

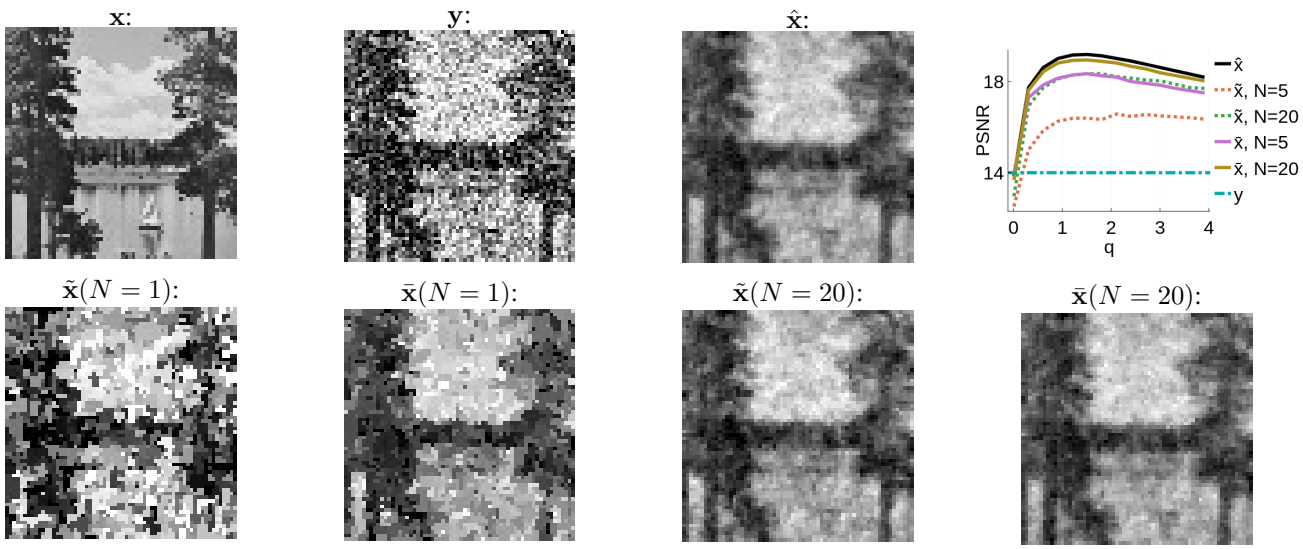

Fig. 1. Illustration on an image. A grayscale image $\mathbf{x}$ is considered as a graph signal on an unweighted 2D grid graph where each pixel is a node connected to its four immediate neighbours. $\mathbf{y} = \mathbf{x} + \boldsymbol{\epsilon}$ is a noisy measurement of $\mathbf{x}$ ($\boldsymbol{\epsilon}$ is Gaussian with covariance matrix $\gamma^2\mathsf{I}$). $\hat{\mathbf{x}} = q\,(q\mathsf{I} + \mathsf{L})^{-1}\,\mathbf{y}$ is the exact Tikhonov denoised signal (here with $q = 1$) that we try to estimate. Bottom line: the two left images show estimates of $\hat{\mathbf{x}}$ obtained with the RSF-based estimators $\tilde{\mathbf{x}}$ and $\bar{\mathbf{x}}$ detailed in Section 3. Averaging over $N = 20$ forest realisations, one obtains the two images on the right. Finally, the top-right figure is the Peak Signal-to-Noise Ratio (PSNR) of the denoised images (averaged over 100 realisations of $\boldsymbol{\epsilon}$). As usual in these scenarios, there exists an optimal regularization parameter (a value of q maximizing the PSNR) that is here observed to be between 1 and 2.

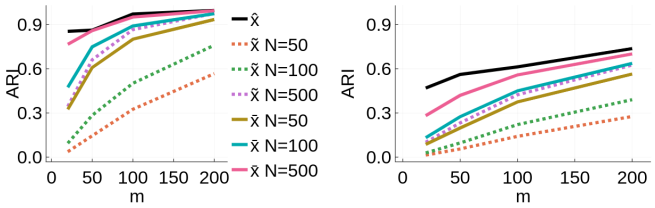

Fig. 2. Illustration on SSL. Performance of the community recovery in a SBM with two equal-size classes, vs. the number of pre-labeled nodes m in each class. Results are averaged over 10 realisations of SBM. Left: setting with strong communities. Right: setting with fuzzy communities.

re-written, with $\mathsf{K} = (\mathsf{Q} + \mathsf{L})^{-1}\,\mathsf{Q}$ and $\mathsf{Q} = \frac{\mu}{2}\mathsf{D}$, as:

$$\forall l = 1, \ldots, k \qquad \mathbf{f}_l = \mathsf{D}^{1-\sigma}\mathsf{K}\mathsf{D}^{\sigma-1}\mathbf{y}_l.$$

Finally, once all classification functions $\mathbf{f}_l$ are computed, each node i is classified in the class $\text{argmax}_l\ f_l(i)$.

One may use our estimators to solve the SSL classification task: $\forall l$, i/ use the proposed estimators on the vector $\mathsf{D}^{\sigma-1}\mathbf{y}_l$ to estimate $\mathsf{K}\mathsf{D}^{\sigma-1}\mathbf{y}_l$, ii/ left-multiply the result by $\mathsf{D}^{1-\sigma}$ and obtain an estimate of $\mathbf{f}_l$, iii/ once all functions $\mathbf{f}_l$ have been estimated, classify each node i to $\text{argmax}_l\ f_l(i)$. In the following, we choose $\sigma = 0$ and set $\mu = 1$.

We illustrate this on the Stochastic Block Model (SBM): a random graph model with communities. Consider a SBM with $n = 3000$ nodes and two communities of equal size. We generate two scenarios: a well-structured (resp. fuzzy) setting with probability of intra-block connection $p_{\text{in}} = 2 \cdot 10^{-2}$ (resp. 10^{-2}) and inter-block connection $p_{\text{out}} = 3 \cdot 10^{-3}$, which corresponds to a sparse graph with average degree $\simeq 35$ (resp. 20). The following experiment is performed: i/ choose m nodes randomly in each community to serve as *a priori* knowledge on the labels and use them to define the two label functions $\mathbf{y}_l$; ii/ compute the two classification functions $\mathbf{f}_l$ either via direct computation (method referred to as $\hat{\mathbf{x}}$) or via our proposed estimators; iii/ each node i is classified in community $\text{argmax}\,[f_1(i), f_2(i)]$. The performance of the community recovery is measured by the Adjusted Rand Index (ARI), a number between -1 and 1: the larger it is, the more accurate the recovery of the ground truth. Results are shown in Fig. 2. The estimator $\bar{\mathbf{x}}$ matches the performance of $\hat{\mathbf{x}}$ after $N = 500$ forest realizations. Also, the smaller the amount of prior knowledge m and the fuzzier the block structure, the harder it is to match $\hat{\mathbf{x}}$. A closer look at the sampled forests shows that some trees do not contain any labeled nodes, thus failing to propagate the label information. This proof-of-concept could be improved in various ways to avoid this difficulty –going beyond the scope of this paper.

5. CONCLUSION

We provide an original and scalable method enabling to estimate graph smoothing operations in large graphs. In future work, we will further explore this deep link between RSFs and Laplacian-based numerical linear algebra, to apply these ideas to more advanced graph signal processing operations.

6. REFERENCES

[1] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *Signal Processing Magazine, IEEE*, vol. 30, no. 3, pp. 83–98, May 2013.

[2] A. Sandryhaila and J.M.F. Moura, "Big Data Analysis with Signal Processing on Graphs: Representation and processing of massive data sets with irregular structure," *Signal Processing Magazine, IEEE*, vol. 31, no. 5, pp. 80–90, Sept. 2014.

[3] Konstantin Avrachenkov, Alexey Mishenin, Paulo Gonçalves, and Marina Sokol, "Generalized Optimization Framework for Graph-based Semi-supervised Learning," in *Proceedings of the 2012 SIAM International Conference on Data Mining*, pp. 966–974.

[4] Gilles Puy, Nicolas Tremblay, Rémi Gribonval, and Pierre Vandergheynst, "Random sampling of bandlimited signals on graphs," *Applied and Computational Harmonic Analysis*, pp. –, 2016.

[5] Yousef Saad, *Iterative methods for sparse linear systems*, vol. 82, SIAM, 2003.

[6] D.I. Shuman, P. Vandergheynst, and P. Frossard, "Chebyshev polynomial approximation for distributed signal processing," in *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, June 2011, pp. 1–8.

[7] F.R.K. Chung, *Spectral graph theory*, Number 92. Amer Mathematical Society, 1997.

[8] Rafig Agaev and Pavel Chebotarev, "Spanning Forests of a Digraph and Their Applications," *Automation and Remote Control*, vol. 62, no. 3, pp. 443–466, 2001, arXiv: math/0602061.

[9] Simon Barthelmé, Nicolas Tremblay, Alexandre Gaudillière, Luca Avena, and Pierre-Olivier Amblard, "Estimating the inverse trace using random forests on graphs," in *Proc. GRETSI Symposium Signal and Image Processing, Lille, France*, 2019, arXiv: 1905.02086.

[10] David Bruce Wilson, "Generating random spanning trees more quickly than the cover time," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 296–303, ACM.

[11] L. Avena and A. Gaudillière, "Two Applications of Random Spanning Forests," *Journal of Theoretical Probability*, July 2017.

[12] Luca Avena and Alexandre Gaudillière, "Random spanning forests, Markov matrix spectra and well distributed points," *arXiv:1310.1723 [math]*, Oct. 2013, arXiv: 1310.1723.

[13] Nicolas Tremblay, Paulo Gonçalves, and Pierre Borgnat, "Chapter 11 - Design of Graph Filters and Filterbanks," in *Cooperative and Graph Signal Processing*, Petar M. Djurić and Cédric Richard, Eds., pp. 299 – 324. Academic Press, 2018.