

# Estimating a graph's spectrum via random Kirchhoff forests

Simon BARTHELMÉ<sup>1</sup> Fabienne CASTELL<sup>2</sup> Alexandre GAUDILLIÈRE<sup>3</sup> Clothilde MELOT<sup>2</sup> Matteo QUATTROPANI<sup>4</sup>  
Nicolas TREMBLAY<sup>1,5</sup>

<sup>1</sup>CNRS, Univ Grenoble-Alpes, Grenoble-INP, GIPSA-lab, Grenoble, France

<sup>2</sup>Aix-Marseille Univ, I2M, Marseille, France

<sup>3</sup>CNRS & Aix-Marseille Univ, I2M, Marseille, France

<sup>4</sup>Department of Mathematics & Physics, Università di Roma Tre, Rome, Italy

<sup>5</sup>Department of Mathematics & Statistics, UiT the Arctic University of Norway, Tromsø, Norway

**Résumé** – Pour des matrices de grande taille, la décomposition spectrale exacte est trop coûteuse, et on ne peut calculer toutes les valeurs propres. Un objectif plus modeste consiste à approcher la loi empirique des valeurs propres, pour connaître approximativement leur répartition. Les approches classiques utilisent des estimateurs de Monte Carlo pour obtenir une estimation des moments de la loi des valeurs propres. Dans cet article, nous introduisons une nouvelle approche pour ce problème, basée sur des forêts aléatoires sur graphes dites forêts de Kirchhoff. Nous montrons comment certaines observables de ces forêts peuvent être exploitées pour obtenir des estimateurs de la densité spectrale empirique de laplaciens de grands graphes. Si la précision souhaitée de l'estimation n'est pas trop importante, notre approche ouvre la voie à des estimations spectrales de graphes en temps sous-linéaire en le nombre de liens.

**Abstract** – Exact eigendecomposition of large matrices is very expensive, and it is practically impossible to compute exact eigenvalues. Instead, one may set a more modest goal of approaching the empirical distribution of the eigenvalues, recovering the overall shape of the eigenspectrum. Current approaches to spectral estimation typically work with *moments* of the spectral distribution. These moments are first estimated using Monte Carlo trace estimators, then the estimates are combined to approximate the spectral density. In this article we show how *Kirchhoff forests*, which are random forests on graphs, can be used to estimate certain non-linear moments of very large graph Laplacians. We show how to combine these moments into an estimate of the spectral density. If the estimate's desired precision isn't too high, our approach paves the way to the estimation of a graph's spectrum in time sublinear in the number of links.

## 1 Introduction

Computing the exact eigenvalue decomposition of a large symmetric matrix  $\mathbf{L} \in \mathbb{R}^{n \times n}$  has cost  $\mathcal{O}(n^3)$  in general, and becomes quickly out of reach for matrices with  $n > 10^4$ . In some cases, all that is needed is an estimate of the  $p$  largest or smallest eigenvalues, and this can be achieved using a variety of Krylov subspace methods [15]. In other cases, what is required is an estimate of the distribution of *all* eigenvalues, an important quantity in various applications [17].

Specifically, letting  $\lambda_1 \dots \lambda_n$  stand for the eigenvalues of  $\mathbf{L}$ , the goal is to estimate :

$$c(\tau) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(\lambda_i \leq \tau) \quad (1)$$

for all  $\tau$ . Here  $c(\tau)$  measures the fraction of eigenvalues that are smaller than the threshold  $\tau$ .

In what follows, it is useful to think of the function  $c(\tau)$  as the cumulative density function of the empirical distribution of the eigenvalues, i.e. the discrete measure

$$\mu = \frac{1}{n} \sum_i \delta_{\lambda_i}. \quad (2)$$

To estimate the c.d.f.  $c(\tau)$ , the standard approach is to first estimate *moments* of the empirical distribution (the *estimation* step) then to reconstruct the cumulative density function from the estimated moments (the *reconstruction* step).

In the existing literature, the estimation step takes advantage of the fact that moments of the empirical measure  $\mu$  correspond to traces of powers of  $\mathbf{L}$  :

$$\mu(\lambda^j) \stackrel{\text{def}}{=} \int \lambda^j d\mu(x) = \frac{1}{n} \sum_i \lambda_i^j = \frac{1}{n} \text{Tr}(\mathbf{L}^j) \quad (3)$$

Computing  $\text{Tr}(\mathbf{L}^j)$  is expensive in general, but stochastic trace estimators [17, 11] like the Hutchinson-Girard estimator can be used. For any random vector  $\mathbf{z}$  with  $\mathbb{E}(\mathbf{z}) = 0$  and  $\mathbb{E}(\mathbf{z}\mathbf{z}^t) = \mathbf{I}$ , we have :

$$\mathbb{E}(\mathbf{z}^t \mathbf{L}^j \mathbf{z}) = \text{Tr}(\mathbf{L}^j) \quad (4)$$

and in this way traces can be computed by taking successive matrix-vector products starting from a random vector  $\mathbf{z}$ .

In the reconstruction step, one tries to reconstruct the cdf of  $\mu$  from knowledge of its  $p$  first moments. Various techniques are available, the most effective are Jackson-Chebyshev expansions [6] and the stochastic Lanczos quadrature [4]. The cost of these techniques scales with the cost of a matrix-vector product  $\mathbf{L}\mathbf{z}$ , which itself scales with  $|\mathcal{E}|$ , the number of edges of the graph. For the very sparse systems that arise from discretising Partial Differential Equations, they are very effective, but in denser systems they can be quite costly.

In what follows we describe a graph-theoretic approach for this problem that is based on a specific random process called the “Kirchhoff forest”. Kirchhoff forests are a certain kind of random forests on graphs, with a range of applications in randomised linear algebra and graph signal processing [13, 8, 2]. We follow the same idea of estimating expectations under the empirical distribution  $\mu$ , followed by a reconstruction step. However, our method is conceptually different from the literature : instead on relying on the computation of quadratic forms  $\mathbf{z}^t \mathbf{L}^j \mathbf{z}$  which prevents existing methods to improve the  $|\mathcal{E}|$  term in the total computation cost, our method is based on sampling random forests of the graph and, under certain assumptions, scales in  $\mathcal{O}(n)$  rather than  $|\mathcal{E}|$ . The downside of this is that, at this stage, our method is only competitive if one only needs a medium-accuracy estimation of the graph's spectrum.

Section 2 introduces Kirchhoff forests and how they can be used to estimate certain spectral expectations. Section 3 outlines the reconstruction method. Section 4 gives some numerical results illustrating how our method outperforms the state-of-the-art if one only requires medium accuracy.

Work partially funded by the ANR project GRANOLA (ANR-21-CE48-0009) and the European Union (Marie Curie project Rand4TrustPool, 101148828).

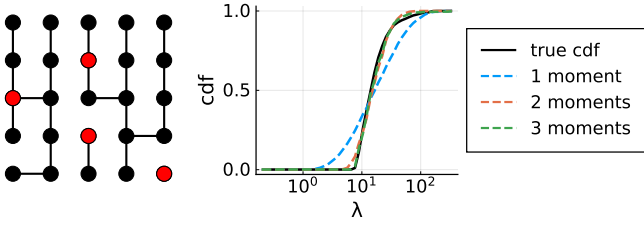


FIGURE 1 : (Left) Illustration of a rooted spanning forest on the 5x5 grid graph, containing 4 trees, the smallest of which is a trivial tree of size 1. Each tree contains a distinguished node called the root. Kirchhoff Forests (KFs) produce random rooted spanning forests with distribution given by eq. (7). (Right) Illustration of a cdf estimation of a graph's spectrum, using our KF-based algorithm (for  $l = 1, 2, 3$  moments), on a Barabasi-Albert graph, with  $n = 10^3$  and  $\bar{d} = 20$ .

## 2 Forests for moment estimation

In what follows we assume that  $\mathbf{L}$  is the Laplacian matrix of a weighted, undirected graph  $G = (\mathcal{V}, \mathcal{E})$  with  $n$  nodes and  $|\mathcal{E}|$  edges :

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \quad (5)$$

where  $\mathbf{A}$  is the adjacency matrix and  $\mathbf{D}$  the degree matrix. See [3] for how to extend this to diagonally-dominant  $\mathbf{L}$ . We denote by  $\mu(f)$  the expectation of function  $f$  under the empirical spectral measure :

$$\mu(f) = \frac{1}{n} \sum_i f(\lambda_i) \quad (6)$$

Kirchhoff forests are random spanning forests on graphs that are particularly easy to sample, and enable efficient estimators of various quantities tied to the graph Laplacian [3, 13, 9].

In the context of graph theory, a *spanning forest* is a subgraph of  $G$ , made up of disjoint trees (acyclic subgraphs), and such that every node belongs to the forest (the spanning part). The concept is illustrated on fig. 1 (left).

A Kirchhoff Forest (KF) is a distribution on *rooted* spanning forests. A forest is said to be rooted if every tree has a distinguished node called the root (see fig. 1). If the tree contains a single node, then the root of the tree is the node itself. KFs are defined by the following probability mass function :

$$p_q(\phi) = \frac{|R(\phi)|^q}{Z_q} \prod_{ij \in \mathcal{E}(\phi)} w_{ij} \quad (7)$$

Here  $\phi$  is a forest with edge set  $\mathcal{E}(\phi)$  and roots  $R(\phi)$ .  $Z_q$  is a normalisation constant, and  $q$  is a parameter that governs the average number of roots in the random forest. Large values of  $q$  favor a large number of roots. In fact, as shown in [1] :

$$\mathbb{E}(|R(\phi)|) = \sum_{i=1}^n \frac{q}{q + \lambda_i} = n \mu(q/(q + \cdot)) \quad (8)$$

[3] used this relationship to define inverse trace estimators. Here we go further and show how to estimate various non-linear expectations under  $\mu$ . KFs have the advantage of being easy to sample. A basic way to sample a KF is via an adaptation of Wilson's algorithm for Uniform Spanning Trees (see [13]). The algorithm constructs trees branch-by-branch by successive aggregation of loop-erased random walks. Wilson's algorithm for a KF with parameter  $q$  runs in time  $\mathcal{O}\left(\frac{|\mathcal{E}|}{q}\right)$ . In our simulations, we use a more sophisticated version, due to [1], the so-called Coupled Forests (CF) algorithm. The CF algorithm produces a continuous sequence of random variables  $\phi_q$ , for  $q \in [q_{min}, q_{max}]$ , where each  $\phi_q$  is a KF with parameter  $q$ . These random variables are coupled (non-independent) but generating the whole family can be done in time  $\mathcal{O}\left(\frac{|\mathcal{E}|}{q_{min}}\right)$ , the same as generating

the most expensive member. Unfortunately, the CF algorithm is too complex to be described fully here. We refer the reader to [1] and to our implementation in *KirchhoffForests.jl*<sup>1</sup>. Sampling such a CF is the main building block of our proposed algorithm, and we will typically do so for  $q_{min}$  set to a fraction of the average degree of the graph  $q_{min} = \bar{d}/\alpha$ , requiring a total  $n$ -linear computation cost of  $\mathcal{O}(\alpha n)$ .

Eq (8) shows that certain observables of KFs are spectral expectations of the form  $\mu(f)$ . Our idea is to estimate a large set of such expectations, specifically,

$$h(q, k) = \mu\left(\left(\frac{q}{q + \cdot}\right)^k\right) \quad (9)$$

for  $q \in [q_{min}, q_{max}]$ ,  $k \in 1, 2, \dots, l$ . In practice we use a discrete grid of values of  $q$  and  $l$  is small (typically  $l = 3$  in our experiments). All these expectations yield information about  $\mu$ , and can be used for reconstruction.

To estimate these quantities, we use the following fact : define

$$\mathbf{K}_q = q(q\mathbf{I} + \mathbf{L})^{-1}; h(q, k) = \frac{1}{n} \text{Tr}(\mathbf{K}_q^k) \quad (10)$$

and for a given KF  $\phi_q$ , the following matrix :

$$\mathbf{M}(\phi_q) = [\mathbb{I}(r_{\phi_q}(i) = j)]_{i \in \mathcal{V}, j \in \mathcal{V}} \quad (11)$$

where  $r_{\phi_q}(i)$  is the root of node  $i$  in the forest (the root of the tree  $i$  belongs to). Authors in [1] have shown that

$$\mathbb{E}(\mathbf{M}(\phi_q)) = \mathbf{K}_q. \quad (12)$$

In particular this implies eq. (8), which gives an unbiased estimator for  $h(q, 1)$  (simply count the number of roots). To generalise, we use the following lemma :

**Lemma 2.1.** *Let  $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(k)}$  random i.i.d. matrices with expectation  $\mathbb{E}(\mathbf{X}) = \mathbf{M}$ . Then*

$$\mathbb{E}\left(\text{Tr}\left(\mathbf{X}^{(1)} \dots \mathbf{X}^{(k)}\right)\right) = \text{Tr}\left(\mathbf{M}^k\right) \quad (13)$$

*Proof.*

$$\begin{aligned} \mathbb{E}\left(\text{Tr}\left(\mathbf{X}^{(1)} \dots \mathbf{X}^{(k)}\right)\right) &= \mathbb{E}\left(\sum_{i=1}^n \sum_{j_1 \dots j_k} \mathbf{X}_{i,j_1}^{(1)} \mathbf{X}_{j_1,j_2}^{(2)} \dots \mathbf{X}_{j_{k-1},i}^{(k)}\right) \\ &= \sum_{i=1}^n \sum_{j_1 \dots j_k} \mathbf{M}(i, j_1) \dots \mathbf{M}(j_{k-1}, i) = \text{Tr}(\mathbf{M}^k). \quad \square \end{aligned}$$

Applying this lemma to  $\mathbf{M}(\phi_q)$  defined in eq. (11), we obtain :

**Theorem 2.2.** *Let  $\phi_q^{(1)} \dots \phi_q^{(k)}$  be  $k$  independent KFs with parameter  $q$ . Let  $\rho_k = r_{\phi_q^{(k)}} \circ r_{\phi_q^{(k-1)}} \dots \circ r_{\phi_q^{(1)}}$  the composition of the root maps, and  $\tilde{h}(q, k) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(\rho_k(i) = i)$ . Then :*

$$\mathbb{E}\left(\tilde{h}(q, k)\right) = \frac{1}{n} \text{Tr}(\mathbf{K}_q^k) = h(q, k) \quad (14)$$

Given  $k$  realisations of the KF  $\phi_q$ ,  $\tilde{h}(q, k)$  is computable at cost  $\mathcal{O}(kn)$ . In addition, one can show that  $\text{Var}(\tilde{h}(q, k)) \leq h(q, k)/n$ , thus ensuring fast convergence of the estimation.

Our overall framework is as follows : pick a logarithmic grid for  $q$ , of size  $n_\lambda$ , ranging from  $q_{min} = \bar{d}/\alpha$  ( $\alpha = 100$  in our experiments) to  $q_{max} = 2d_{max}$ . Recall that  $\bar{d}$  (resp.  $d_{max}$ ) refers to the mean (resp. maximum) degree of the graph. Note that  $2d_{max}$  is a classical upper bound for  $\lambda_{max}$ , obtained from applying the Gershgorin circle theorem to  $\mathbf{L}$ . We run  $l$  iterations of the Coupled Forests algorithm, to obtain a first estimate of  $h(q, k)$  for  $k$  in  $1 \dots l$  at each value of  $q$  in the grid. We replicate the whole process  $s$  times to obtain several Monte-Carlo estimates that we finally average. At the end of the day, we obtain (very) precise estimates of  $h(q, k)$  for all values of  $q$  in the  $n_\lambda$ -sized grid and for  $k \in [1 \dots l]$ , for an overall computation cost of  $\mathcal{O}((\alpha + n_\lambda)sl n)$ . Several avenues are open to further improve the variance of these estimates, by adapting techniques from [13, 14] for instance. We now move on to explaining how these estimates can be used for reconstruction of the spectral cdf  $c(\tau)$ .

<sup>1</sup>available here : [github.com/dahtah/KirchhoffForests.jl](https://github.com/dahtah/KirchhoffForests.jl)

### 3 Reconstruction

At the end of the first phase, we have collected estimates of  $h(q, k)$  for several values of  $q$  and  $k$ . All of these contain some information about  $\mu$  the spectral density of  $\mathbf{L}$ , but extracting and combining that information is not trivial. We will outline a procedure that we call “fixed- $q$ ” estimation, which gives good results in practice. However, we do not wish to claim that it is in any way optimal; the question of optimally combining the results of the estimation phase is unsolved as yet. The starting point of the fixed- $q$  procedure consists in noticing that if we define :

$$x_i = \frac{q}{q + \lambda_i} \quad (15)$$

then :

$$h(q, k) = \frac{1}{n} \sum \left( \frac{q}{q + \lambda_i} \right)^k = \frac{1}{n} \sum x_i^k \quad (16)$$

which are the classical moments of the empirical distribution of the  $x_i$ ’s. We note this measure

$$\nu_q = \frac{1}{n} \sum_i \delta_{x_i} \quad (17)$$

Since we have access to the moments of  $\nu_q$ , we can apply the standard toolset of classical moment theory to form an approximation. In our numerical simulations we rely on a maximum entropy estimator, defined below. This allows us to form an estimator  $\hat{\nu}_q$  based on the estimated moments  $\hat{h}(q, 1) \dots \hat{h}(q, l)$ .

The map from  $\lambda$  to  $x$  given by 15 is nonlinear, and it sends all the eigenvalues that are small (compared to  $q$ ) to  $x \approx 1$ , and all those that are very large to  $x \approx 0$ . In theory, the map is invertible and we could recover an estimate of  $\mu$  from  $\hat{\nu}_q$  by a change of variables. However, we only have access to a few (estimated) moments of  $\nu_q$  and the change of variables may be unstable. What we do instead is simply form a pointwise estimate at the point where the map 15 has highest slope (in magnitude) : for  $\lambda = q$ . Noting that the cdf in  $q$  is  $c(q) = \mu([0, q]) = 1 - \nu_q([0, \frac{1}{2}])$ , we set

$$\tilde{c}(q) = \tilde{\nu}_q \left( \left[ 0, \frac{1}{2} \right] \right).$$

We collect these estimates for all values of  $q$  in the grid, yielding an estimate of the overall cdf  $c(\tau)$ . An example is shown in fig. 1 (right).

**Maximum entropy estimator.** Let us now explain how  $\nu_q$  can be approximated from knowledge of its  $k$  first moments. Our approach is inspired by the maximum-entropy principle, a widely-used heuristic for problems of this type [12]. The maximum entropy principle suggests that when all that is known about a certain distribution are a fixed number of moments, then a reasonable estimate of the unknown distribution is the distribution with maximum entropy that has compatible moments. Importantly, for fixed  $q$ , we have a lower bound on  $x_i$ ,  $x_i \geq \frac{q}{q + \lambda_{\max}} \geq \frac{q}{q + 2d_{\max}}$  (see eq. (15)). This implies that the distribution  $\nu_q$  has support on  $[\frac{q}{q + 2d_{\max}}, 1]$ , and we can use this information to constrain our problem further.

The maximum-entropy distribution with support on  $[a, b]$  and moments  $\mathbf{m} = [m_1 \dots m_l]^t$  is given by :

$$\underset{\nu \in \mathcal{P}_{[a, b]}(\mathbf{m})}{\operatorname{argmin}} KL(\nu, \nu_0) \quad (18)$$

where  $KL(\nu, \nu_0)$  is the KL divergence with respect to the uniform measure on  $[a, b]$ , and  $\mathcal{P}_{[a, b]}(\mathbf{m})$  is the set of probability measures on  $[a, b]$  compatible with the moment constraints (i.e. such that  $\nu(x^j) = m_j$  for  $j \in 1 \dots l$ ). As is well-known ([5], ch. 12), the solution of eq. (18) belongs to an exponential family of the form :

$$q_{\beta}(x) = \exp(\beta^t \mathbf{v}(x) - \psi(\beta)) \quad (19)$$

where  $\mathbf{v}(x) = [x^i]_{i=1}^l$  and  $\psi(\beta) = \log \int_a^b \exp(\beta^t \mathbf{v}(x)) dx$ . The optimal  $\beta$  can be found by solving :

$$\beta^* = \underset{\beta \in \mathbb{R}^s}{\operatorname{argmin}} \psi(\beta) - \beta^t \mathbf{m} \quad (20)$$

Given the solution  $\beta_*$ , we can estimate  $\nu([a, \frac{1}{2}])$  as  $\int_a^{\frac{1}{2}} \exp(\beta_*^t \mathbf{v}(x) - \psi(\beta_*)) dx$ .

Now, there are a number of statistical and numerical issues to deal with. First of all, computing  $\psi(\beta)$  involves an intractable integral that we approximate by Gauss-Legendre quadrature ([7], ch. 6). More importantly, the moments are only known approximately. We define a confidence ellipsoid  $B$  of the form :

$$B = \left\{ \mathbf{m}' \in \mathbb{R}^l \text{ such that } \sum_{i=1}^l v_i^{-1} (m'_i - m_i)^2 \leq 1 \right\}$$

where  $v_i$  is an upper-bound on the variance of the estimate  $m_i$ . When solving eq. 20, we interrupt the optimisation once the moments corresponding to the iterate  $\beta_t$  are in  $B$ . In addition, a vector of estimated moments  $\mathbf{m}$  can fall outside the set of valid moments  $\mathcal{M}_l([a, b])$  [18]. For instance, for all probability distributions, we always have  $\nu(x^2) \geq \nu(x)^2$ , but estimated moments may have  $m_2 < m_1$  because of the noise. To deal with this, we proceed as follows :

1. We first test that the estimated moments are in  $\mathcal{M}_l([a, b])$ . If they are, we proceed with the maximum entropy estimate.

2. Otherwise, we project the estimated moments  $\mathbf{m}$  on  $\mathcal{M}_l([a, b])$  (which is a form of denoising). We then apply maximum entropy to the denoised moments.

Step 1 involves checking classical admissibility conditions for truncated moment sequences, as described for instance in [16], p. 230. It involves checking that two specific matrices formed from  $\mathbf{m}$  are positive-definite. The denoising in Step 2 can be carried out via a reformulation as a semidefinite program, too involved to be described here, see [18] or [10].

### 4 Numerical experiments

Fig. 1 (right) illustrates a spectrum cdf estimation for a realization of a random Barabasi-Albert graph, with our method. We observe how the estimation improves as  $l$  increases. Let us now compare our method to the state-of-the-art. The three methods we compare are :

- **poly** : a polynomial approximation method with Jackson-Chebyshev damping coefficients, as in [6]. **Moment estimation step** : estimate the first  $p$  moments of  $L$  by averaging  $\{\mathbf{x}^t \mathbf{L}^i \mathbf{x}\}_{i \in [1..p]}$  over  $r$  random vectors  $\mathbf{x}$  at an overall cost of  $\mathcal{O}(rp|\mathcal{E}|)$  ( $|\mathcal{E}|$  is equal to the number of non-zero entries of  $\mathbf{L}$ ). **Reconstruction step** : for each value of the grid over  $\lambda$  on which the cdf estimation is to be made, a precise linear combination of these estimates is computed, based on the Jackson-Chebyshev approximation of the step function.

- **slq**, short for Stochastic Lanczos Quadrature (see for instance [4]). **Moment estimation<sup>2</sup> step** : for  $r$  independent realizations of a random vector  $\mathbf{x}$ , this method starts by creating a  $p \times p$  tri-diagonal Lanczos matrix based on  $\mathbf{x}$  and its  $p$  successive left-multiplication by  $\mathbf{L}$ . This is done with an overall cost of  $\mathcal{O}(rp|\mathcal{E}|)$ . **Reconstruction step** : these  $r$  matrices are then diagonalized and results averaged.

- **forests** : our method. **Moment estimation step** : the moments  $\{h(q, k)\}_{k=1..l}$  for  $n_{\lambda}$  values of  $q$  spaced between  $\bar{d}/\alpha$  and  $2d_{\max}$  are estimated for an overall cost of  $\mathcal{O}(\alpha l n)$  to sample the  $l$  coupled forest +  $\mathcal{O}(n_{\lambda} l n)$  for the computation of the  $l$ -th order roots at each  $q$  of the grid of size  $n_{\lambda}$ . This is repeated  $s$  times to obtain Monte-Carlo averages, yielding an overall cost of  $\mathcal{O}((\alpha + n_{\lambda}) s l n)$ . **Reconstruction step** : as described in Section 3.

Note that the reconstruction cost of each method is independent of  $n$ , and thus negligible asymptotically in  $n$  : in the following, we do not take into account its computation cost as we want to show trends in large  $n$ . Also, note that in the above,  $s$  and  $r$  are similar “Monte-Carlo parameters”, and  $p$  and  $l$  are similar “moment-order parameters”. For different sizes  $n$  of graphs, different types of graphs, and the parameter choices  $\alpha = 100$  and  $n_{\lambda} = 15$ , we plot in Fig. 2

<sup>2</sup>not truly a moment estimation step, but close enough to keep this term

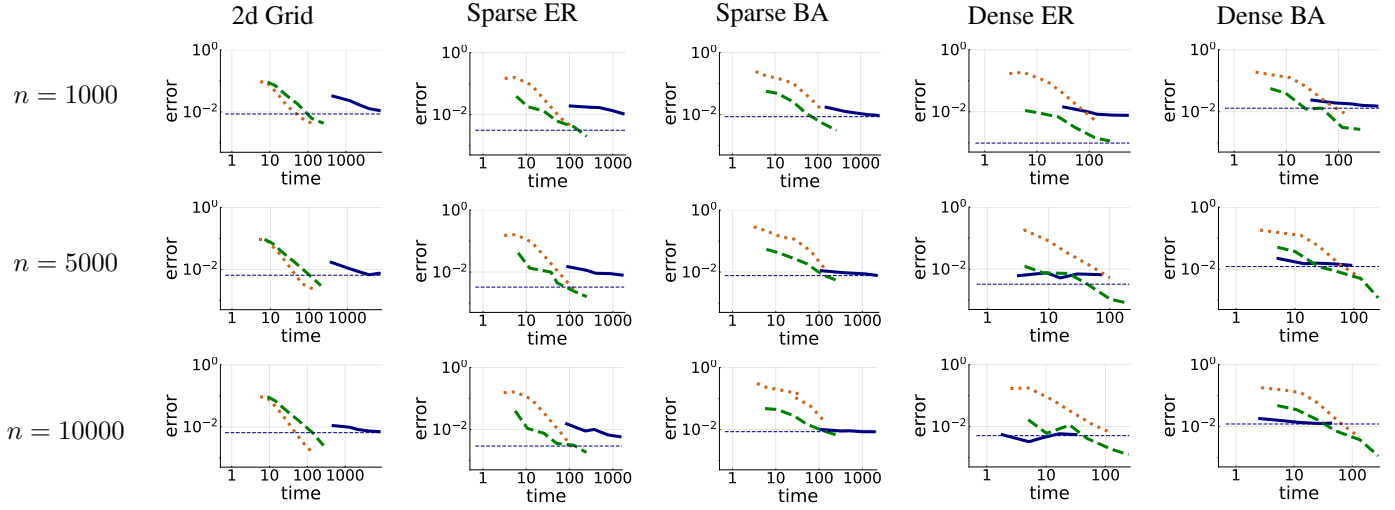


FIGURE 2 : **Estimation error versus computation time** for 3 types of methods : `forests` (ours, in solid blue), `poly` (dotted orange), and `slq` (dashed green). The dotted horizontal blue line is the result of our reconstruction algorithm if we feed it the exact moments (rather than the KF-estimated ones). Each column is for a different type of graph : ER stands for Erdős-Renyi, BA for Barabasi-Albert, “sparse” means average degree of 20, and “dense” means average degree of  $n/10$ . Each line is for a different value of the number of nodes in the graph. For each graph and each value of  $n$ , the time axis is normalized by the time of computation of the corresponding matrix-vector multiplication  $Lx$ . Results are averaged over 50 realizations of all three methods (that are all stochastic), and 10 realizations of each graph (for the 4 random graphs of the list).

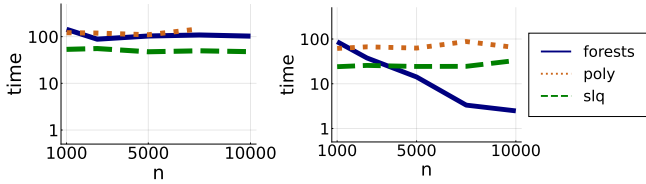


FIGURE 3 : **Computation time required to reach a 2% error versus the number of nodes**, for all three methods. Left : sparse BA graph. Right : dense BA graph. The time axis is normalized by the time of computation of the matrix-vector multiplication  $Lx$ .

the reconstruction error of the cdf versus the computation time of each method. For `poly` and `slq`, we show the results for  $r = 5$  and  $p \in [1, 2, 5, 10, 25, 50]$  (each value of  $p$  giving a data point in the error-computation time axes of the figure) as we have observed in our examples that performance stabilizes quickly in  $r$ . Regarding our method, it would seem sensible to fix the Monte-Carlo parameter  $s$  as above and make the moment parameter  $l$  vary. However, we are confronted with an important limitation of our method as it is currently : whereas the moment estimation step could work with increasing  $l$ 's, the reconstruction step starts to show signs of instability<sup>3</sup> for  $l \geq 4$ . We thus only plot results for  $l = 3$  and  $s \in [1, 3, 5, 10, 20]$ .

Even with this small, fixed value of  $l$ , we observe that if one accepts a moderate precision in the estimation, then our method can provide significant performance improvements. Note that this happens even though the error reconstruction is necessarily lower bounded – here by the horizontal blue lines – as opposed to the other two methods for which the error decreases continuously with  $p$ . For instance, if one accepts a 2% reconstruction error, Fig. 3 shows the time needed to reach that error versus the number of nodes for sparse and dense graphs. As expected from the previous theoretical time analyses of each method, the time to reach a given error is linear in the number of edges<sup>4</sup>  $|\mathcal{E}|$  for `poly` and `slq` and *sublinear* in  $|\mathcal{E}|$  (in fact, linear in  $n$ ) for our method – which is, we believe, remarkable.

## 5 Concluding remarks

This is a first communication on a new and promising class of methods to estimate a graph's spectrum. Whereas state-of-the-art methods are classically based on successive matrix-vector multiplications  $Lx$

(and as such cannot pretend to go beyond a linear cost in  $|\mathcal{E}|$ ), our results, by subtly subsampling the graph in several random spanning forests, pave the way to sublinear-time (in  $|\mathcal{E}|$ ) spectrum estimation. We experimentally confirm this at least when moderate precision is enough, and when both the lowest  $\lambda$  for which one wants an estimate of the cdf (in our experiments, we chose  $\bar{d}/\alpha$  with  $\alpha = 100$ ) and the grid-size chosen for the spectrum's estimate,  $n_\lambda$  (equal to 15 in our experiments), are independent of  $n$ . Current work in progress is devoted to stabilizing the reconstruction algorithm and go beyond  $l = 3$  in order to increase our method's precision.

## References

- [1] L. Avena and A. Gaudillièr. Two Applications of Random Spanning Forests, *Journal of Theoretical Probability*, 2017.
- [2] L. Avena, F. Castell, A. Gaudillièr, and C. Mélot. Intertwining wavelets or multiresolution analysis on graphs through random forests, *Applied and Computational Harmonic Analysis*, 2020.
- [3] Barthelmé, Tremblay, Gaudillièr, Avena and Amblard. Estimating the inverse trace using random forests on graphs, *GRETSI*, 2019.
- [4] T. Chen, T. Trogon and S. Ubaru. Analysis of stochastic Lanczos quadrature for spectrum approximation. *ICML Proceedings*, 2021.
- [5] T. Cover and J. Thomas. Elements of Information Theory. Wiley, 2005.
- [6] E. di Napoli, E. Polizzi and Y. Saad. Efficient estimation of eigenvalue counts in an interval. *Numerical Linear Algebra with Applications*, 2016.
- [7] G. Golub and G. Meurant. Matrices, moments and quadrature with applications. *Princeton University Press*, 2009.
- [8] Jaquard, Amblard, Barthelmé, Tremblay. Random Multi-Type Spanning Forests for Synchronization on Sparse Graphs. *arXiv :2403.19300*, 2024.
- [9] Jaquard, Fanuel, Amblard, Bardenet, Barthelmé, Tremblay. Smoothing Complex-Valued Signals on Graphs with Monte-Carlo. *ICASSP*, 2023.
- [10] J-B. Lasserre. A semidefinite programming approach to the generalized problem of moments. *Mathematical Programming*, 2008.
- [11] P-G. Martinsson and J. Tropp. Randomized numerical linear algebra : Foundations and algorithms. *Acta Numerica*, 2020.
- [12] L. Mead. Maximum entropy in the problem of moments. *Journal of Mathematical Physics*, 1984.
- [13] Y. Pilavci, P-O. Amblard, S. Barthelmé and N. Tremblay. Graph tikhonov regularization and interpolation via random spanning forests. *IEEE transactions on Signal and Information Processing over Networks*, 2021.
- [14] Pilavci, Amblard, Barthelmé and Tremblay. Variance Reduction for Inverse Trace Estimation via Random Spanning Forests. *GRETSI*, 2022.
- [15] Y. Saad. Numerical methods for large eigenvalue problems. *SIAM*, 2011.
- [16] K. Schmüdgen. The Moment Problem. *Springer*, 2017.
- [17] A. Weiße, G. Wellein, A. Alvermann and H. Fehske. The kernel polynomial method. *Reviews of modern physics*, 2006.
- [18] Y. Wu and P. Yang. Optimal estimation of Gaussian mixtures via denoised method of moments. *The Annals of Statistics*, 2020.

<sup>3</sup>inherited from well-known instabilities of maximum-entropy algorithms

<sup>4</sup>The time to compute  $Lx$  is linear in  $|\mathcal{E}|$