ACCELERATED SPECTRAL CLUSTERING USING GRAPH FILTERING OF RANDOM SIGNALS

Nicolas Tremblay$^{(1,2,3)}$*, Gilles Puy*$^{(1)}$*, Pierre Borgnat*$^{(2)}$*, Rémi Gribonval*$^{(1)}$ *and Pierre Vandergheynst*$^{(1,3)}$

$^{(1)}$ INRIA Rennes - Bretagne Atlantique, Beaulieu Campus, Rennes, France
$^{(2)}$ Physics Laboratory, ENS Lyon, CNRS, University of Lyon, Lyon, France
$^{(3)}$ Institute of Electrical Engineering, EPFL, Lausanne, Switzerland

ABSTRACT

We build upon recent advances in graph signal processing to propose a faster spectral clustering algorithm. Indeed, classical spectral clustering is based on the computation of the first k eigenvectors of the similarity matrix' Laplacian, whose computation cost, even for sparse matrices, becomes prohibitive for large datasets. We show that we can estimate the spectral clustering distance matrix without computing these eigenvectors: by graph filtering random signals. Also, we take advantage of the stochasticity of these random vectors to estimate the number of clusters k. We compare our method to classical spectral clustering on synthetic data, and show that it reaches equal performance while being faster by a factor at least two for large datasets.

Index Terms— Spectral clustering, graph signal processing, graph filtering

1. INTRODUCTION

Spectral clustering has become a popular clustering algorithm, due to its simplicity of implementation and high performance for many different types of datasets [2, 3]. Given a set of N data points $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$, it basically transforms them non-linearly into a k-dimensional space first, by computing a similarity matrix $\boldsymbol{W}$ from the data, and second by computing the k first eigenvectors of its Laplacian. Calculating these eigenvectors is the computational bottleneck of spectral clustering: it becomes prohibitive when N becomes large and/or when the k-th eigengap becomes too small [2]. Circumventing this issue is an active area of research [4, 5]. Another difficulty, common to all clustering methods, is the estimation of the usually unknown number of clusters k [6, 7].

We propose a new method that avoids the Laplacian's partial diagonalization and proposes a stability measure to estimate k. This method is based on the emerging field of graph signal processing [8, 9], where the graph we consider here is defined by the weighted adjacency matrix $\boldsymbol{W}$. In our previous work [10], we proposed the use of the graph wavelet [11] or scaling function [12] transforms of random vectors to detect multiscale communities in networks. In this paper, we build upon this idea of using filtered random vectors as probes of the underlying graph's structure; and further prove that ideal low-pass graph filters have deep connections with spectral clustering. Taking advantage of the fast graph filtering defined in [13] to low-pass filter such random signals *without computing the first k eigenvectors of the graph's Laplacian*, we propose an accelerated spectral clustering method that has the collateral advantage of defining a stability measure to estimate the number of clusters k.

In Section 2, we recall graph signal processing notations and tools used in the paper, as well as the classical spectral clustering algorithm. In Section 3, we prove that one can filter random signals to estimate the spectral clustering distance matrix. In Section 4, we detail our algorithm and propose a stability measure to estimate the number of clusters k. We finally show results obtained on a controled dataset in Section 5; before concluding in Section 6.

2. BACKGROUND

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ be an undirected weighted graph with $\mathcal{V}$ the set of N nodes, $\mathcal{E}$ the set of edges, and $\mathbf{W}$ the weighted adjacency matrix such that $W_{ij} = W_{ji} \geq 0$.

2.1. The graph Fourier matrix

Consider the graph's combinatorial Laplacian[1] matrix $\boldsymbol{L} = \mathbf{S} - \mathbf{W}$ where $\mathbf{S}$ is diagonal with $S_{ii} = s_i = \sum_{j\neq i} W_{ij}$ the strength of node i. $\boldsymbol{L}$ is real symmetric, therefore diagonalizable in an orthogonal basis: its spectrum is composed of its set of eigenvalues $\{\lambda_l\}_{l=1\dots N}$ that we sort: $0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_N$; and of $\boldsymbol{\chi}$ the orthonormal matrix of its eigenvectors: $\boldsymbol{\chi} = (\boldsymbol{\chi}_1|\boldsymbol{\chi}_2|\dots|\boldsymbol{\chi}_N)$. Considering only connected graphs, the multiplicity of eigenvalue $\lambda_1 = 0$ is

This work was partly funded by the European Research Council, PLEASE project (ERC-StG-2011-277906); and the ANR-14-CE27-0001 GRAPHSIP grant. This paper's general results were recently and independently published by another team in [1].

[1] One could use other types of Laplacians, such as the normalized Laplacian; which adds a normalization step [14] after step 2 of the spectral clustering algorithm (see Section 2.2) and slightly changes our subsequent proofs.

one [15]. By analogy to the continuous Laplacian operator whose eigenfunctions are the classical Fourier modes and eigenvalues their squared frequencies, the columns of $\boldsymbol{\chi}$ are considered as the graph's Fourier modes, and $\{\sqrt{\lambda_l}\}_l$ as its set of associated "frequencies" [9]. Other types of graph Fourier matrices have been proposed (e.g. [16]), but in order to exhibit the link between graph signal processing and classical spectral clustering (that partially diagonalizes the Laplacian matrix), the Laplacian-based Fourier matrix is more natural.

2.2. Spectral clustering

Let us recall the method of spectral clustering [17, 2]. The input is the set of data points $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$ and k the number of clusters one desires. Follow the steps:

1. Compute the pairwise similarities $s(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and create a similarity[2] graph $\boldsymbol{W}$. Compute its Laplacian $\boldsymbol{L}$.
2. Let $\boldsymbol{U} \in \mathbb{R}^{N\times k}$ contain $\boldsymbol{L}$'s first k eigenvectors: $\boldsymbol{U} = (\boldsymbol{\chi_1}|\boldsymbol{\chi_2}|\cdots|\boldsymbol{\chi_k})$. In other words, the columns of $\boldsymbol{U}$ are the first k low-frequency Fourier modes of the graph.
3. Treat each node i as a point in $\mathbb{R}^k$ by defining its feature vector $\boldsymbol{f}_i \in \mathbb{R}^k$ as the i-th row of $\boldsymbol{U}$:
$$\boldsymbol{f}_i = \boldsymbol{U}^\top \boldsymbol{\delta}_i, \tag{1}$$
where $\delta_i(j) = 1$ if $j = i$ and 0 elsewhere.
4. Run k-means (or any clustering algorithm) with the Euclidean distance $D_{ij} = ||\boldsymbol{f}_i - \boldsymbol{f}_j||$ to obtain k clusters.

2.3. Graph filtering

The graph Fourier transform $\hat{\boldsymbol{x}}$ of a signal $\boldsymbol{x}$ defined on the nodes of the graph reads: $\hat{\boldsymbol{x}} = \boldsymbol{\chi}^\top \boldsymbol{x}$. Given a continuous filter function h defined on $[0, \lambda_N]$, its associated graph filter operator $\boldsymbol{H} \in \mathbb{R}^{N\times N}$ is defined as: $\boldsymbol{H} = \boldsymbol{\chi}\widehat{\boldsymbol{H}}\boldsymbol{\chi}^\top$, where $\widehat{\boldsymbol{H}} = \text{diag}(h(\lambda_1), h(\lambda_2), \cdots, h(\lambda_N))$. We write $\boldsymbol{Hx}$ the signal $\boldsymbol{x}$ filtered by h. In the following, we will consider ideal low-pass filters, generically noted h_{λ_c}, such that:

$$h_{\lambda_c}(\lambda) = 1 \text{ if } \lambda \leq \lambda_c \quad \text{and} \quad h_{\lambda_c}(\lambda) = 0 \text{ if not,} \tag{2}$$

and $\boldsymbol{H}_{\lambda_c}$ its associated graph filter operator.

2.4. Fast graph filtering

In order to filter a signal by h without diagonalizing the Laplacian matrix, one may rely [13] on a polynomial approximation of order m of h on $[0, \lambda_N]$: $\exists\{\alpha_l\}_{l\in[0,m]}$ s.t. $h(\lambda) \simeq \sum_{l=0}^m \alpha_l \lambda^l$. This enables us to approximate $\boldsymbol{Hx}$:

$$\boldsymbol{Hx} = \boldsymbol{\chi}\widehat{\boldsymbol{H}}\boldsymbol{\chi}^\top \boldsymbol{x} \simeq \sum_{l=0}^m \alpha_l \boldsymbol{L}^l \boldsymbol{x} = \mathcal{F}_h^m \boldsymbol{x}, \tag{3}$$

[2]See [2] for several choices of similarity measure s as well as several ways to create $\boldsymbol{W}$ from the $s(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

where we note that the approximation only requires matrix-vector multiplication and effectively bypasses the diagonalisation of the Laplacian; and where we introduce the notation $\mathcal{F}_h^m$ to design the fast filtering operator of order m associated to h. This fast filtering method has a total complexity of $O(m(|\mathcal{E}| + N))$ [13], which in the case of sparse graph where $|\mathcal{E}| \sim N$, ends up being $O(mN)$; compared to the $O(N^3)$ complexity needed to diagonalize the Laplacian matrix.

Note on the choice of m. The lower we choose m, the faster is the computation, but the less precise is the polynomial approximation. Fixing a maximal error of approximation to δ, we note m^* the minimal value of $m \in \mathbb{N}^*$ such that:

$$\sup_{\lambda\in[0,\lambda_N]} \left| h(\lambda) - \sum_{l=0}^m \alpha_l \boldsymbol{L}^l \boldsymbol{x} \right| \leq \delta. \tag{4}$$

In this paper, we only consider ideal low-pass h_{λ_c} defined on $[0, \lambda_N]$. Notice that $h_{\lambda_c/\lambda_N} = h_{\lambda_c}(\lambda\lambda_N)$ is always defined on $[0, 1]$ and does not depend on any graph anymore. One can therefore *beforehand* tabulate m^* as a function of its sole parameter λ_c/λ_N (and δ). Then, given a real filter h_{λ_c} defined on $[0, \lambda_N]$ to approximate, one only needs to refer to this table and choose $m = m^*(\lambda_c/\lambda_N; \delta)$. Here, we fix $\delta = 0.1$.

3. ACCELERATING SPECTRAL CLUSTERING

3.1. Filtering random signals to estimate D_{ij}

We show that we can estimate the distance $D_{ij} = ||\boldsymbol{f}_i - \boldsymbol{f}_j||$ by filtering a few random signals with the ideal low-pass h_{λ_k} (as defined in Eq. (2) with $\lambda_c = \lambda_k$). First of all, consider the matrix operator $\boldsymbol{H}_{\lambda_k}$ associated to h_{λ_k}. One may write:

$$\boldsymbol{H}_{\lambda_k} = \boldsymbol{\chi} \begin{pmatrix} \boldsymbol{I}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \boldsymbol{\chi}^\top = \boldsymbol{U}\boldsymbol{U}^\top, \tag{5}$$

where $\boldsymbol{I}_k$ is the identity of size k, and $\boldsymbol{0}$ null block matrices.

Second, consider the matrix $\boldsymbol{R} = (\boldsymbol{r}_1|\boldsymbol{r}_2|\cdots|\boldsymbol{r}_\eta) \in \mathbb{R}^{N\times\eta}$ consisting of η random signals $\boldsymbol{r}_i$, whose components are independent random Gaussian variables of zero mean and variance $1/\eta$. Consider its filtered version $\boldsymbol{H}_{\lambda_k}\boldsymbol{R} \in \mathbb{R}^{N\times\eta}$, and define node i's feature vector $\tilde{\boldsymbol{f}}_i \in \mathbb{R}^\eta$ as the i-th line of this filtered matrix:

$$\tilde{\boldsymbol{f}}_i = (\boldsymbol{H}_{\lambda_k}\boldsymbol{R})^\top \boldsymbol{\delta}_i. \tag{6}$$

Proposition: *Let $\epsilon, \beta > 0$ be given. If η is larger than:*

$$\eta_0 = \frac{4 + 2\beta}{\epsilon^2/2 - \epsilon^3/3} \log N, \tag{7}$$

then with probability at least $1 - N^{-\beta}$, we have: $\forall(i,j) \in [1,N]^2$

$$(1-\epsilon)||\boldsymbol{f}_i - \boldsymbol{f}_j||^2 \leq ||\tilde{\boldsymbol{f}}_i - \tilde{\boldsymbol{f}}_j||^2 \leq (1+\epsilon)||\boldsymbol{f}_i - \boldsymbol{f}_j||^2. \tag{8}$$

Proof. We rewrite $||\tilde{\boldsymbol{f}}_i - \tilde{\boldsymbol{f}}_j||^2$ in a form that will let us apply the Johnson-Lindenstrauss lemma of norm conservation:

$$\begin{aligned} ||\tilde{\boldsymbol{f}}_i - \tilde{\boldsymbol{f}}_j||^2 &= ||\boldsymbol{R}^\top \boldsymbol{H}_{\lambda_k}^\top (\boldsymbol{\delta}_i - \boldsymbol{\delta}_j)||^2 \\ &= ||\boldsymbol{R}^\top \boldsymbol{U}\boldsymbol{U}^\top (\boldsymbol{\delta}_i - \boldsymbol{\delta}_j)||^2 \\ &= ||\boldsymbol{R}^\top \boldsymbol{U}(\boldsymbol{f}_i - \boldsymbol{f}_j)||^2. \end{aligned} \tag{9}$$

As i) the columns of $\boldsymbol{U}$ are normalized to 1 and orthogonal to each other, and ii) $\boldsymbol{R}$ is a random Gaussian matrix with mean zero and variance $1/\eta$, then $\boldsymbol{R}' = \boldsymbol{R}^\top \boldsymbol{U}$ is also Gaussian with same mean and variance; and Equation (9) reads:

$$||\tilde{\boldsymbol{f}}_i - \tilde{\boldsymbol{f}}_j||^2 = ||\boldsymbol{R}'(\boldsymbol{f}_i - \boldsymbol{f}_j)||^2. \tag{10}$$

This enables us to apply Theorem 1.1 of [18] (an instance of the Johnson-Lindenstrauss lemma) and finish the proof. □

Consequence: Setting β to 1, and therefore the failure probability of Equation (8) to $1/N$, we only need to filter $\eta \gtrsim \frac{12}{\epsilon^2}\log N$ random signals to estimate (up to an error ϵ) the spectral clustering distance matrix. How this error ϵ on the distance estimation theoretically affects the performance of the spectral clustering algorithm is still, to our knowledge, an open question. We observe experimentally (see Sec. 5) that using a number $\eta \gtrsim k$ (i.e. allowing a relatively high error $\epsilon^2 \simeq 12\frac{\log N}{k}$) is usually enough for satisfying performance.

3.2. Fast filtering of random signals

In practice, we do not exactly filter these random signals by h_{λ_k} as the computation of $\boldsymbol{H}_{\lambda_k}$ requires the diagonalisation of the Laplacian, which is precisely what we are trying to avoid. Instead, we take advantage of the fast filtering scheme recalled in Section 2.4. Still, one question remains: the fast filtering is based on the polynomial approximation of h_{λ_k}, which is itself parametrized by λ_k. Unless we compute the first k eigenvectors of $\boldsymbol{L}$, thereby loosing our efficiency edge on other methods, we cannot know *exactly* the value of λ_k.

To circumvent this, we estimate the spectrum's cumulative density function as in Section VB of [19] (the *cdf* of a graph's spectrum is roughly linear only if the graph shows topological regularity; if not, it is not trivial). Given a graph Laplacian $\boldsymbol{L}$, and a "cutting frequency" λ_c, the algorithm estimates the number of eigenvalues inferior to λ_c. Using a dichotomous procedure on $[0, \lambda_N]$, we obtain an estimate of λ_k, noted $\tilde{\lambda}_k$.

4. ACCELERATED SPECTRAL CLUSTERING

4.1. Algorithm

Consider a set of data points $(\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_N})$ and k the number of desired clusters. The first step of the algorithm does not change as compared to Section 2.2: compute the pairwise similarities s_{ij}, create a similarity graph $\boldsymbol{W}$, and compute its Laplacian $\boldsymbol{L}$. Then:

1. Estimate [20] $\boldsymbol{L}$'s largest eigenvalue λ_N (necessary for steps 2 and 3).
2. Estimate $\boldsymbol{L}$'s k-th eigenvalue $\tilde{\lambda}_k$ as in Sec. 3.2.
3. Set $m = m^*(\tilde{\lambda}_k/\lambda_N; \delta)$ as in Sec. 2.4; compute the polynomial approximation of order m of the ideal low-pass filter $h_{\tilde{\lambda}_k}$ to obtain the fast filtering operator $\mathcal{F}^m_{\tilde{\lambda}_k}$.
4. Generate η random Gaussian signals of mean 0 and variance $1/\eta$: $\boldsymbol{R} = (\boldsymbol{r}_1|\boldsymbol{r}_2|\cdots|\boldsymbol{r}_\eta) \in \mathbb{R}^{N\times\eta}$.
5. Filter these signals with $\mathcal{F}^m_{\tilde{\lambda}_k}$ and define, for each node i, its feature vector $\tilde{\boldsymbol{f}}_i \in \mathbb{R}^\eta$:
$$\tilde{\boldsymbol{f}}_i^\top = \boldsymbol{\delta}_i^\top \mathcal{F}^m_{\tilde{\lambda}_k} \boldsymbol{R}. \tag{11}$$
6. Run k-means (or any clustering algorithm) with the Euclidean distance $\tilde{D}_{ij} = ||\tilde{\boldsymbol{f}}_i - \tilde{\boldsymbol{f}}_j||$ to obtain k clusters.

4.2. Complexity considerations

We compare the time complexity of this algorithm and the classical spectral clustering algorithm. Let us separate our algorithm (resp. the classical algorithm) into two parts: the spectral estimation part consisting of steps 1 and 2 (resp. step 2) and the clustering part consisting in steps 4 to 6 (resp. steps 3 and 4). According to [21], k-means has complexity $O(\eta N)$ where η is the dimension of the N considered vectors. Considering the discussion of Section 2.4, the time complexity of the clustering part is thus $O(\eta(m+1)N)$ (resp. $O(kN)$). The time complexity of the spectral estimation part is difficult to estimate, as it depends [2] on the graph-dependent eigengap $\lambda_{k+1} - \lambda_k$. In both algorithms, the larger is this eigengap, the faster is the convergence. Nevertheless, empirical observations show that, even though the clustering part of the classical algorithm is faster than ours; our algorithm makes up to that difference by computing even faster (especially for large N) its spectral estimation part than the classical one.

4.3. Estimate the number of clusters k

In real data, the number of clusters k to find is usually unknown. Instead, one has access to a (possibly large) interval of values $[k_{min}, k_{max}]$. Notions of stability to estimate k have been used in various contexts [7, 22]; and we propose here a new one that naturally comes from the random vectors' stochasticity. For all $k \in [k_{min}, k_{max}]$, we perform our algorithm J times using J different realisations of the η random signals, to obtain J different clusterings $\{\mathcal{C}_k^j\}_{j\in[1,J]}$. We define a stability measure as the mean of the Adjusted Rand Index similarity [23] between all pairs of clusterings [10, 11]:

$$\gamma(k) = \frac{2}{J(J-1)} \sum_{(i,j)\in[1,J]^2, i\neq j} \texttt{ari}(\mathcal{C}_k^i, \mathcal{C}_k^j). \tag{12}$$

Denote k^* the value of k for which γ reaches its global maximum: we consider it as the relevant number of clusters.

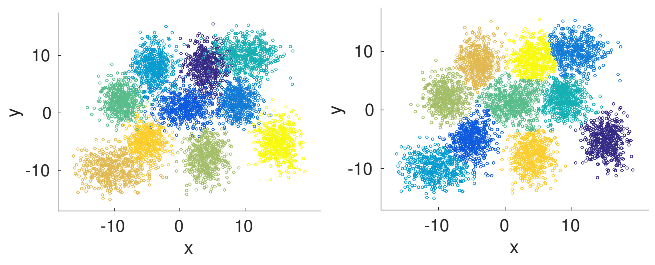

Fig. 1. (left) One realisation of Gaussian synthetic data with $N = 5000$ and $k = 10$. Colours indicate the label of each point. (right) Recovered labeling with our method and $\eta = 2k$ (recovery performance $p = 0.85$). A similar result is obtained using the original spectral clustering algorithm.

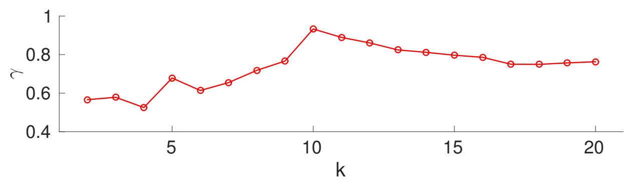

Fig. 2. Stability measure γ versus the number of clusters k for the Gaussian dataset illustrated in Fig. 1: its maximum correctly detects $k = 10$. $N = 5000, \eta = 2k$ and $J = 20$.

5. RESULTS

We consider a sum of $k = 10$ two-dimensional Gaussians with different means and variances, and from this distribution, draw a set of N points in $\mathbb{R}^2$. Each point i has a label $l(i)$ indicating from which Gaussian it was drawn. Fig. 1 (left) shows a realisation of such distribution with $N = 5000$ where colours indicate the label of each node. The goal here is to recover the original labeling l. To measure how close the output l_m of a given method recovers the original labeling, we compute the Adjusted Rand Index $p(l_m) = \texttt{ari}(l, l_m) \in [-1, 1]$: the closer is the performance p to 1, the closer is l_m to l.

From these N points, we create a similarity matrix by building a K nearest neighbours graph with $K \sim \log N$ as suggested in [2] as a classical way of generating sparse similarity graphs. Other wiring possibilities exist to create such a graph (see [2]) but we only show this particular one as the choice of construction does not affect our results (but still considering graphs with same sparsity).

We use Matlab's k-means function (with 20 replicates) and the GSP toolbox [24] for steps 2 and 3 of our algorithm.

First, we remark in this dataset that values of $\lambda_{k=10}/\lambda_N$ are small (of the order of 10^{-4}), which necessitates a large $m \sim 200$ for a correct approximation of the ideal low-pass on $[0, \lambda_N]$ (see Section 2.4).

Let us now illustrate in Fig. 2 the stability measure γ obtained using our method with $N = 5000$, $\eta = 2k$ and $J = 20$: the global maximum correctly detects $k = 10$ and Fig. 1 (right) shows one recovered labeling for $k = 10$.

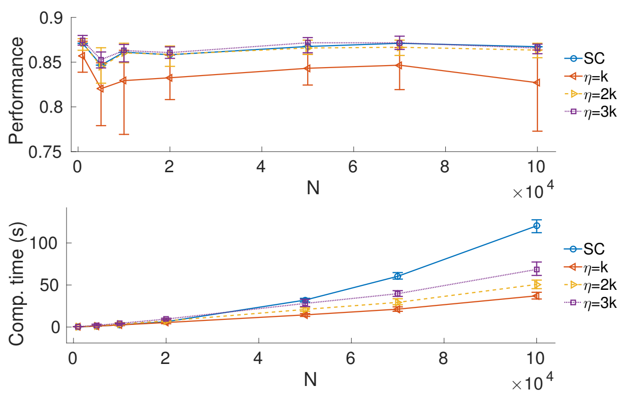

Fig. 3. Average (and 10 and 90% quantiles) performance (top) and computation time (bottom) over 20 realisations of the Gaussian dataset illustrated in Fig. 1 for the classical spectral clustering algorithm (SC), and our proposition for $\eta = k, 2k$ and $3k$.

We compare our method vs. classical spectral clustering in Figure 3 in terms of performance and time of computation. We note that for $\eta = 2k$ and $\eta = 3k$, our method performs as well as the classical algorithm; while for $\eta = k$, on the other hand, the number of random signals becomes insufficient as we observe the recovery starting to fail. Up to $N \simeq 4.10^4$, the computation time is slightly faster with the classical algorithm. But as N increases, the classical algorithm's computing time increases significantly faster than our proposition's: for $N = 10^5$ for instance, computation time is 2 (resp. 2.5, 3) times faster when one uses $\eta = 3k$ (resp. $2k$, k) random signals than the classical algorithm.

6. CONCLUSION

We propose a new method that paves the way to alternative spectral clustering methods bypassing the usual computational bottleneck of extracting the Laplacian's first k eigenvectors. We take advantage of the fast graph low-pass graph filtering of a few random vectors to estimate the spectral clustering distance. The use of random vectors makes our algorithm stochastic, which in turn enables us to define a stability measure γ for any k: scales of interest maximize γ. Results on synthetic data show that our method is scalable and for $\eta \gtrsim k$, one has the same performance as with the classical spectral algorithm while reducing the time complexity by a few factors.

We prooved that the error on the estimation of the distance D_{ij} is well controlled, but the question of how such an error propagates on the estimation of the clusters themselves is open. Moreover, the impact of the error δ of the polynomial approximation on the rest of the algorithm is still largely unknown and matter of future work.

7. REFERENCES

[1] D. Ramasamy and U. Madhow, "Compressive spectral embedding: sidestepping the SVD," in *Advances in Neural Information Processing Systems*, Dec 2015, pp. 550–558.

[2] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[3] H. Jia, S. Ding, X. Xu, and R. Nie, "The latest research progress on spectral clustering," *Neural Computing and Applications*, vol. 24, no. 7-8, pp. 1477–1486, 2014.

[4] J. Lei and A. Rinaldo, "Consistency of spectral clustering in stochastic block models," *Ann. Statist.*, vol. 43, no. 1, pp. 215–237, 02 2015.

[5] U. Kang, B. Meeder, E.E. Papalexakis, and C. Faloutsos, "Heigen: Spectral analysis for billion-scale graphs," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 2, pp. 350–362, Feb 2014.

[6] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sept. 1999.

[7] S. Ben-David, U. von Luxburg, and D. Pl, "A sober look at clustering stability," in *Learning Theory*, Gbor Lugosi and HansUlrich Simon, Eds., vol. 4005 of *Lecture Notes in Computer Science*, pp. 5–19. Springer Berlin Heidelberg, 2006.

[8] 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.

[9] 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.

[10] N. Tremblay and P. Borgnat, "Graph wavelets for multiscale community mining," *Signal Processing, IEEE Transactions on*, vol. 62, no. 20, pp. 5227–5239, Oct 2014.

[11] N. Tremblay and P. Borgnat, "Multiscale community mining in networks using the graph wavelet transform of random vectors," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, Dec 2013, pp. 463–466.

[12] S. Roux, N. Tremblay, P. Borgnat, P. Abry, H. Wendt, and P. Messier, "Multiscale anisotropic texture unsupervised clustering for photographic paper," in *7th IEEE International Workshop on Information Forensics and Security (WIFS)*, 2015.

[13] 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.

[14] A.Y. Ng, M.I. Jordan, Y. Weiss, et al., "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

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

[16] A. Sandryhaila and J.M.F. Moura, "Discrete signal processing on graphs," *Signal Processing, IEEE Transactions on*, vol. 61, no. 7, pp. 1644–1656, April 2013.

[17] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.

[18] D. Achlioptas, "Database-friendly random projections: Johnson-lindenstrauss with binary coins," *Journal of Computer and System Sciences*, vol. 66, no. 4, pp. 671 – 687, 2003, Special Issue on {PODS} 2001.

[19] D.I. Shuman, C. Wiesmeyr, N. Holighaus, and P. Vandergheynst, "Spectrum-adapted tight graph wavelet and vertex-frequency frames," *Signal Processing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[20] L.N. Trefethen and D. Bau III, *Numerical linear algebra*, vol. 50, Siam, 1997.

[21] R. Xu and D. Wunsch II, "Survey of clustering algorithms," *Neural Networks, IEEE Transactions on*, vol. 16, no. 3, pp. 645–678, May 2005.

[22] A. Ben-Hur, A. Elisseeff, and I. Guyon, "A stability based method for discovering structure in clustered data.," in *Pacific symposium on biocomputing*. World Scientific, 2002, vol. 7, pp. 6–17.

[23] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[24] N. Perraudin, J. Paratte, D. Shuman, V. Kalofolias, P. Vandergheynst, and D.K. Hammond, "Gspbox: A toolbox for signal processing on graphs," *arXiv preprint arXiv:1408.5781*, 2014.