

Approximating the Spectrum of a Graph

David Cohen-Steiner

INRIA Sophia-Antipolis
david.cohen-steiner@inria.fr

Christian Sohler

Dept. of Computer Science, TU Dortmund
christian.sohler@tu-dortmund.de

Weihao Kong

Dept. Computer Science, Stanford University
whkong@stanford.edu

Gregory Valiant

Dept. Computer Science, Stanford University
gvaliant@cs.stanford.edu

ABSTRACT

The spectrum of a network or graph $G = (V, E)$ with adjacency matrix A , consists of the eigenvalues of the normalized Laplacian $L = I - D^{-1/2}AD^{-1/2}$. This set of eigenvalues encapsulates many aspects of the structure of the graph, including the extent to which the graph possesses community structures at multiple scales. We study the problem of approximating the spectrum, $\lambda = (\lambda_1, \dots, \lambda_{|V|})$, of G in the regime where the graph is too large to explicitly calculate the spectrum. We present a sublinear time algorithm that, given the ability to query a random node in the graph and select a random neighbor of a given node, computes a succinct representation of an approximation $\tilde{\lambda} = (\tilde{\lambda}_1, \dots, \tilde{\lambda}_{|V|})$, such that $\|\tilde{\lambda} - \lambda\|_1 \leq \epsilon|V|$. Our algorithm has query complexity and running time $\exp(O(1/\epsilon))$, which is *independent* of the size of the graph, $|V|$. We demonstrate the practical viability of our algorithm on synthetically generated graphs, and on 15 different real-world graphs from the Stanford Large Network Dataset Collection, including social networks, academic collaboration graphs, and road networks. For the smallest of these graphs, we are able to validate the accuracy of our algorithm by explicitly calculating the true spectrum; for the larger graphs, such a calculation is computationally prohibitive. The spectra of these real-world networks reveal insights into the structural similarities and differences between them, illustrating the potential value of our algorithm for efficiently approximating the spectrum of large large networks.

KEYWORDS

Spectral graph theory, sublinear algorithms, random walks, method of moments.

ACM Reference Format:

David Cohen-Steiner, Weihao Kong, Christian Sohler, and Gregory Valiant. 2018. Approximating the Spectrum of a Graph. In *KDD 2018: 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Christian acknowledges the support of ERC grant 307696 and of the German Science Foundation, collaborative research center 876, project A6. David's contribution is supported by ERC grant 339025 $\hat{\Delta}$ IJGUDHIA $\hat{\Delta}$ L. Weihao's and Gregory's contributions are supported by NSF award CCF-1704417 and by ONR award N00014-17-1-2562.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD 2018, August 19–23, 2018, London, United Kingdom

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5552-0/18/08...\$15.00

<https://doi.org/10.1145/3219819.3220119>

August 19–23, 2018, London, United Kingdom. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3219819.3220119>

1 INTRODUCTION

Given an undirected graph $G = (V, E)$, its normalized Laplacian matrix is defined as $L = I - D^{-1/2}AD^{-1/2}$, where D is the diagonal matrix with entries $D_{i,i}$ given by the degree of the i th vertex, and A is the adjacency matrix of the graph. It is not hard to see that L is positive semidefinite and singular, with eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{|V|}$, whose sum is $\text{trace}(L)$. Many structural and combinatorial properties of graphs are exposed by the eigenvalues (and eigenvectors) of the associated graph Laplacian, L . For example, as was quantified in a recent series of works [12, 16, 21], the value of the i th eigenvalue provides insights into the extent to which the graph admits a partitioning into i components. Hence the spectrum provides a detailed sense of the community structures present in the graph at multiple scales. Macroscopic features of the spectrum also provide information about the graph structure—for example, the extent to which the spectrum is symmetric about the middle eigenvalue $\lambda_{|V|/2}$ corresponds to whether the graph tends to be locally bipartite, versus, for example, having many triangles.

Inspecting the spectrum of a graph also serves as a approach to evaluating the plausibility of natural generative models for families of graphs (see, e.g. [6]): for example, if the spectrum of random power-law graphs does not closely resemble the spectrum of the Twitter graph, it suggests that a random power-law graph might be a poor model for the Twitter graph.

Given the structural information contained in the spectrum of a graph's Laplacian, it seems natural to ask the following question: *How much information must one collect about a graph in order to accurately approximate its spectrum?*

1.1 Our results

We give the first sublinear time approximation algorithm for computing the spectrum of a graph $G = (V, E)$. Our algorithm assumes that we can sample vertices uniformly at random from V and that we can also query for a random neighbor of a vertex $v \in V$. (Many graph data structures efficiently support these two operations.) Equivalently, this model corresponds to assuming that we can perform a random walk in G , as well as randomly restart such a walk. For a desired error parameter, $\epsilon \in (0, 1)$, our algorithm performs $\exp(O(1/\epsilon))$ such queries to the graph—independent of the graph size, $|V|$ —then performs a computation taking time $\text{poly}(1/\epsilon)$, and outputs an approximation $\tilde{\lambda}$ of the spectrum λ of the normalized

Laplacian of G (see Definition 3.1 for the formal definition of the normalized Laplacian).

The spectrum of G consists of $|V|$ numbers, and our algorithm will instead output a succinct representation of this spectrum, having size only $O(1/\epsilon)$. We now describe the sense in which this succinct description approximates the true spectrum. Note that the true spectrum can be regarded as a discrete distribution over $[0, 2]$, the “spectral distribution,” which places mass $1/|V|$ at each value λ_i . Our algorithm will output a succinct representation of the spectrum, consisting of only $O(1/\epsilon)$ numbers, which approximates this distribution in the Wasserstein metric. Specifically, the L_1 distance between the cumulative distribution function of the distribution returned by our algorithm, will have ℓ_1 distance at most ϵ from the cumulative distribution function of the true spectral distribution. The representation output by our algorithm corresponds to approximating each of the $1/\epsilon$ quantiles of the spectrum—i.e. an approximation of the $\epsilon|V|$ th smallest eigenvalue, the $2\epsilon|V|$ th smallest, the $3\epsilon|V|$ th smallest, etc. If desired, this succinct representation can then be trivially converted in time $|V|$ into a length $|V|$ vector $\hat{\lambda}_1, \dots, \hat{\lambda}_{|V|}$ such that $\sum_{i=1}^{|V|} |\lambda_i - \hat{\lambda}_i| \leq \epsilon|V|$.

THEOREM 1.1. *For any $\epsilon, \delta > 0$, given the ability to 1) select a uniformly random node from a graph $G = (V, E)$, and 2) for any node, query a uniformly random neighbor of that node, with runtime and number of queries bounded by $\exp(O(1/\epsilon)) \log(1/\delta)$ one can output a succinct representation of the normalized Laplacian of G , which, with probability at least $1 - \delta$, approximates the true spectrum to additive error ϵ in Wasserstein (earth-mover) distance.*

Our algorithm for approximating the spectrum is based on approximating the first $O(1/\epsilon)$ “spectral moments”, the quantities $\frac{1}{|V|} \sum_{i=1}^{|V|} \lambda_i^\ell$ for integers $\ell = 1, 2, \dots$. These moments are traces of matrix powers of the random walk matrix of G , allowing us to approximate them by estimating the return probabilities of length ℓ random walks. Given accurate estimates of the spectral moments, the spectrum can then be recovered by solving a moment-inverse problem, namely recovering a distribution whose moments closely match the estimated spectral moments.

Complementing the strong theoretical result of Theorem 1.1, we empirically demonstrate the practicality of the proposed algorithm, both on a series of synthetically generated graphs, as well as on 15 real-world networks, including road networks, collaboration networks, and social networks. The similarities and differences between the spectra for these networks reveals rich insights into these graphs. We discuss these experiments, and the insights that they reveal, in Section 5.2.

Finally, beyond the general result of Theorem 1.1 which requires a number of walks that is exponential in the inverse of the desired accuracy, ϵ , we also give an algorithm with an improved dependence that applies to planar graphs of bounded degree (such as road networks), and generalizations of planar graphs. This result is mainly of conceptual and theoretical interest, and we did not investigate the empirical properties of the resulting algorithm.

THEOREM 1.2. *For a graph G of maximum degree d that is planar, or that does not contain a forbidden minor, H , one can approximate the spectrum of G to Wasserstein (Kantorovich/earth-mover) distance ϵ with queries and computation time $\left(\frac{d}{\epsilon}\right)^{O(\log(1/\epsilon))}$.*

The proof of this improved result for bounded degree planar graphs requires two tools. The first is the observation that the earth mover distance between the spectra of two graphs is at most twice the graph edit distance (the number of edges that must be added/removed to transform one graph into the other). The second tool is an algorithmic gadget called a “planar partitioning oracle” which allows a planar graph of degree at most d to be partitioned into connected components of size $O(d/\epsilon^2)$, while removing only ϵn edges from the graph. Given such a decomposed graph, the spectrum can then be pieced together from approximations of the spectra of the various pieces.

1.2 Related work

Since the 1970’s, spectral graph theory has flourished and led to the development and understanding of rich connections between structural and combinatorial properties of graphs, and the eigenvalues and eigenvectors of their associated graph Laplacians (see e.g. [3]). From an algorithmic standpoint spectral methods provide useful tools that have been fruitfully employed to solve a number of graph problems including graph coloring, graph searches (e.g. web search), and image partitioning [22, 23]. In terms of the structural interpretations of the eigenvalues, it is easy to see that the multiplicity of the zero eigenvalue is exactly the number of connected components of a graph. Cheeger’s inequality gives a robust analog of this statement, showing a correspondence between the value of the second eigenvalue, and the extent to which the graph can be partitioned into two pieces. Very recently, a series of works [12, 16, 21] developed a “higher order” Cheeger inequality, quantifying a correspondence between the i th eigenvalue and the extent to which the graph admits a partitioning into i components.

There has been a great deal of work characterizing the spectrum of various models of random graphs, including Erdos-Renyi graphs [5], and graphs that attempt to model the properties exhibited by real-world graphs and social networks, including random power-law graphs, small-world graphs, and *scale-free* networks (see e.g. [2, 6]). One way of testing the plausibility of such models is by comparing their spectrum to those of actual real-world networks, though one challenge is the computational difficulty of computing the spectrum for large graphs, which, in the worst case, requires time cubic in the number of nodes of the graph.

Beyond the graph setting, there is a significant body of work from the statistics community on estimating the spectrum of the covariance matrix of a high-dimensional distribution, given access to independent samples from the distribution [1, 4, 9, 14, 15, 19]. As with a graph, the eigenvalues of the covariance matrix of a distribution contain meaningful structural information about the distribution in question, including quantifying the amount of low-dimensional structure. Recently, [10] showed that the spectrum of the covariance of a distribution can be accurately recovered given a number of samples that is sublinear in the dimension, by leveraging a method-of-moments approach that directly estimates the low-order moments of the true spectral distribution. While that work is in the rather different setting of estimating the covariance spectrum of a distribution, and the results have a completely different form than the results of this work (i.e. they require $O(d^{1-\epsilon})$ samples to estimate a d -dimensional distribution to error ϵ), they share the

same high level approach of 1) approximating the spectral moments, and 2) solving the moment-inverse problem to return a vector or distribution whose moments match the approximated moments. The moment estimators in the two works are unrelated, though we can use the same (fairly standard) linear-programming approach to the moment-inverse problem leveraged in that work.

Finally, there is a wealth of work on computing approximate spectral decompositions of matrices (and tensors) in time that is sub-linear in the size of the matrix in question (see e.g. [7, 11, 24, 26, 27]). While superficially similar to our goal, this line of work proceeds by sampling columns (generally either uniformly at random or via importance sampling) from the matrix in question, and the techniques apply to settings where the matrix in question is close to low-rank or has other structure; these approaches are not applicable to our problem of recovering an approximation of the entire spectrum. (For a concrete example, given a graph corresponding to a large rectangular grid, the submatrix generated as an intermediate step in these approaches will have almost all columns completely empty, and will not recover the equally-spaced “linear” spectrum of such a graph.)

1.3 Preliminaries and Notation

Let A be an $n \times n$ real-valued matrix. A value λ is called an *eigenvalue* of A , if there exists a vector v such that $Av = \lambda v$. If A is a symmetric matrix then its eigenvalues and eigenvectors are real. If $A = Q\Lambda Q^{-1}$ where Λ is a diagonal matrix, we say that A has an eigendecomposition. The entries on the diagonal of Λ are the eigenvalues and the columns of Q the eigenvectors of A . If A is symmetric and real-valued it always has an *eigendecomposition* of the form $A = Q\Lambda Q^T$, i.e. Q is an *orthogonal* matrix ($Q^{-1} = Q^T$).

Two matrices A and B are similar, if they can be written as $A = PBP^{-1}$ for an $n \times n$ invertible matrix P . Similar matrices have the same eigenvalues. We may assume w.l.o.g. that the eigenvalues satisfy $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, (where each eigenvalue appears with its algebraic multiplicity) and refer to this sorted list of eigenvalues as the *spectrum*. A matrix is stochastic, if its columns are non-negative reals that sum up to 1.

Throughout, we will also view this list of eigenvalues as a distribution, consisting of n equally-weighted point masses at values $\lambda_1, \dots, \lambda_n$. We refer to this distribution as the *normalized spectral measure* or *spectral distribution*. We will be concerned with recovering this spectral distribution in terms of the *Wasserstein-1* distance metric (i.e. “earth mover distance”). We denote the earth mover distance between two real-valued distributions p and q by $W_1(p, q)$; this distance represents the minimum, over all schemes of “moving” the probability mass of p to yield distribution q , where the cost per unit probability mass of moving from probability x to y is $|x - y|$.

The task of learning the spectral distribution in earth mover distance is closely related to the task of learning the sorted vector of eigenvalues in ℓ_1 distance. This is because the ℓ_1 distance between two sorted vectors of length n is exactly n times the earth mover distance between the corresponding point-mass distributions. Similarly, given a distribution, Q , that is close to the spectral distribution μ_λ in Wasserstein distance, one can transform Q into a length n vector whose ℓ_1 distance is at most $n \cdot W_1(Q, \mu_\lambda)$. (See Lemma 2.5.)

In the remainder of this paper we will assume that A is an $n \times n$ real-valued stochastic matrix with real eigenvalues of absolute value at most 1 and n linearly independent eigenvectors. In particular, we can write $A = Q\Lambda Q^{-1}$. We use e_i to denote the i -th vector of the standard basis of \mathbb{R}^n .

2 APPROXIMATING THE SPECTRUM OF A STOCHASTIC MATRIX

In this section we consider the task of approximating the spectrum of a stochastic matrix, A , given a certain query access to information about A . Our results on estimating the spectrum of a graph Laplacian, which we give in Section 3, will follow from the results of this section, as learning the spectrum of a graph’s Laplacian is equivalent to learning the spectrum of the stochastic matrix corresponding to a random walk on the graph in question.

2.1 Model of computation

To cleanly reason about the properties of the proposed algorithm, it will be useful to formalize the way we can access the matrix in question, and count each such access as taking one unit of time. We will assume that we can access matrix A as follows: Given a number j , we can obtain a value $C(j) \in \{1, \dots, n\}$ distributed according to the j -th column of matrix A . This type of access to A allows us to perform a random walk on A . We note that the time it takes to actually implement such an oracle depends on how the graph is represented in memory. If the graph is stored via adjacency lists then the oracle can be implemented in time $O(\deg)$ per oracle call; if the neighboring vertices are stored as arrays and the node degrees are also stored, this oracle can be implemented in a constant number of arithmetic operations per call. (In our experiments, we represent the graphs in this latter form.)

2.2 Approximating the spectral moments

We proceed via the method of moments: we first obtain accurate estimates of the low-order moments of the spectral distribution, and then leverage these moments to yield the spectral distribution.

Definition 2.1. Let $A = Q\Lambda Q^{-1}$ be a stochastic $n \times n$ matrix with real eigenvalues $1 \geq \lambda_n \geq \dots \geq \lambda_1 \geq -1$. The ℓ -th moment of the spectrum of A is defined as $\frac{1}{n} \sum_{i=1}^n \lambda_i^\ell$.

We will leverage the fact that the trace of a matrix A equals n times the first moment and the trace of $A^i = Q\Lambda^i Q^{-1}$ equals n times the i -th spectral moment, i.e. $\text{Tr}(A^i) = \sum_{i=1}^n \lambda_i^i$.

At the same time, we can also view the trace of A as the sum of return probabilities of a random walk using the transition probabilities of A , i.e.

$$\begin{aligned} \text{Tr}(A^i) &= \sum_{j=1}^n e_j^T A^i e_j \\ &= \sum_{j=1}^n \Pr[i\text{-step Rand. Walk from } j \text{ returns to } j]. \end{aligned}$$

Next we note that we can view

$$\frac{1}{n} \sum_{1 \leq j \leq n} \Pr[i\text{-step Random Walk from } j \text{ returns to } j]$$

as the expected return probability of a random walk starting at j , where j is chosen uniformly at random from $\{1, \dots, n\}$ and transitions probabilities are given by matrix A . Thus, given access to A as described in Section 2.1, the following algorithm can be used as an unbiased estimator for the spectral moments:

```

APPROXSPECTRALMOMENT( $A, \ell, s$ ):
  for  $i = 1$  to  $s$ 
    pick  $j \in \{1, \dots, n\}$  uniformly at random
     $w = j$ 
    for  $k = 1$  to  $\ell$  do
      Let  $w'$  be drawn from the distribution of the
       $w$ -th column of  $A$ 
       $w = w'$ 
    if  $w = j$  then  $X_i = 1$  else  $X_i = 0$ 
  return  $\frac{1}{s} \cdot \sum_{i=1}^s X_i$ 

```

The following lemma follows directly from a Hoeffding bound on the sum of independent 0/1 random variables.

LEMMA 2.2. *For parameters $\epsilon, \delta \in (0, 1)$, fix an integer $s \geq \frac{1}{2}\epsilon^{-2} \ln(2/\delta)$. Given access to the column distributions of a stochastic $n \times n$ matrix $A = Q\Lambda Q^{-1}$ with real eigenvalues $1 \geq \lambda_n, \dots, \geq \lambda_1 \geq -1$, with probability at least $1 - \delta$, algorithm APPROXSPECTRALMOMENT(A, ℓ, s) approximates the ℓ -th spectral moment of A to within an additive error ϵ . Furthermore, the runtime is $O(s\ell)$.*

2.3 Approximating the spectrum from its moments

Given accurate approximations of the first ℓ moments of a distribution, how accurately can one recover the distribution? In the case where the distribution in question is supported on a bounded interval—in our case, supported on the range $[-1, 1]$ as the eigenvalues of a stochastic matrix lie in this interval—the distribution can be recovered to Wasserstein distance (i.e. L_1 distance between the cumulative distribution functions) at most $O(1/\ell)$, plus a term that depends linearly on the magnitude of the approximation errors and exponentially on the number of moments, ℓ . This result follows from the dual definition of Wasserstein distance, as the supremum over Lipschitz-1 functions, f , of the discrepancy in the expected value of f with respect to the two distribution, and the fact that any Lipschitz-1 function can be approximated to L_∞ error $O(1/\ell)$ via a degree ℓ polynomial whose coefficients are bounded in magnitude by an exponential function of ℓ . The following fact, from [25], summarizes this recoverability:

PROPOSITION 2.3 (THEOREM 3 IN [25], SEE ALSO PROP. 1 IN [10]). *Given two distributions with respective density functions p, q supported on $[-1, 1]$ whose first ℓ moments are $\alpha = (\alpha_1, \dots, \alpha_\ell)$ and $\beta = (\beta_1, \dots, \beta_\ell)$, respectively, the Wasserstein distance, $W_1(p, q)$, between p and q is bounded by:*

$$W_1(p, q) \leq \frac{2\pi}{\ell} + 3^\ell \sum_{i=1}^{\ell} |\alpha_i - \beta_i|.$$

The above proposition holds in a worst-case sense—there are certainly sets of moments that robustly determine the distribution to higher accuracy. For example, if the first two moments happen to be 0, then the distribution is fully determined (it is the point mass at

0), and no additional moments are necessary. For many of the real-world networks we considered empirically, our recovery seems to be significantly better than what the theory would suggest—likely due to this sort of phenomena.

Proposition 2.3 asserts that *every* distribution whose first k moments closely match those of the spectral distribution will be a close approximation (in the Wasserstein sense). Hence, algorithmically, one simply needs to find one such distribution. The following classical linear programming approach recovers the distribution (supported on a fine discrete mesh), whose moments match the approximated moments as closely as possible.

MOMENTINVERSE:

Inputs: Vector $\hat{\alpha}$ consisting of the first ℓ approximate moments for a distribution supported on the interval $[a, b]$, and a parameter $\epsilon > 0$.

Output: Distribution p .

- (1) Define $\mathbf{x} = x_0, \dots, x_t$ with $x_i = a + i\epsilon$ and $t = \lceil \frac{b-a}{\epsilon} \rceil$.
- (2) Let $\mathbf{p}^+ = (p_0^+, \dots, p_t^+)$ be the solution to the following linear program, which should be interpreted as a distribution with mass p_i^+ at location x_i :

$$\begin{aligned}
 &\underset{\mathbf{p}}{\text{minimize}} && \sum_{i=1}^{\ell} \left| \hat{\alpha}_i - \sum_{j=0}^t p_j \cdot x_j^i \right| \\
 &\text{subject to} && \sum_{j=0}^t p_j = 1, \text{ and } p_j > 0 \quad \forall j \in \{0, 1, \dots, t\}.
 \end{aligned} \tag{1}$$

- (3) Return the distribution corresponding to \mathbf{p}^+ , namely that assigns mass p_j^+ to location x_j .

Note that the constraints of the linear program in the MomentInverse algorithm—that the entries of \mathbf{p} sum to 1 and are non-negative—ensure that the vector corresponds to a distribution. The objective function asserts that, among distributions supported on values in \mathbf{x} , the returned distribution minimizes sum of the discrepancies between the estimated moment vector, and the moments of the returned distribution.

2.4 Approximating the spectrum of A

We now assemble the above components to characterize the ability to recover the spectral distribution of a stochastic matrix:

THEOREM 2.4. *Given access to the column distributions of a stochastic $n \times n$ matrix $A = Q\Lambda Q^{-1}$ with real eigenvalues $1 \geq \lambda_n, \dots, \lambda_1 \geq -1$, with probability $1 - \delta$ we can approximate the spectrum of A with additive error ϵ in Wasserstein distance with running time and query complexity $e^{O(1/\epsilon)} \log(1/\delta)$.*

PROOF. Throughout, for clarity, we omit all constant factors and use ‘big-O’ notation to indicate that we are suppressing an absolute constant. We will first run Algorithm ApproxSpectralMoment to recover approximations of the first $\ell = O(1/\epsilon)$, and choose $s > \exp(O(1/\epsilon)) \log(2\ell/\delta) = \exp(O(1/\epsilon)) \log(1/\delta)$ such that Lemma 2.2 guarantees that each of the first ℓ spectral moments are recovered to error at most $1/\exp(O(1/\epsilon))$, with failure probability at most δ/ℓ each, and hence via a union bound, the total failure probability across all ℓ moments is at most δ . Let $\hat{\alpha}$ denote

these ℓ approximated moments, and let α denote the true moments. Assume the constant in the $\exp(O(1/\epsilon))$ term of s is chosen such that, with probability at least $1 - \delta$, $\sum_{i=1}^{\ell} |\alpha_i - \hat{\alpha}_i| \ll \epsilon/3^\ell$.

Given these moments, we now analyze the performance of MomentInverse given as input $\hat{\alpha}$, the interval $[-1, 1]$, and the discretization parameter $\epsilon_2 = 1/\exp(O(1/\epsilon))$. We must argue that there exists a good feasible point for the linear program—namely a distribution supported on \mathbf{x} whose moment closely match $\hat{\alpha}$. Given the existence of such a distribution, it follows that the linear program must return a distribution whose moments are at *most* this distance from $\hat{\alpha}$, as the linear program returns the distribution of closes moments. To prove the existence of such a point, consider the true distribution \mathbf{p} whose moments are α , that has been discretized so as to be supported at the ϵ_2 -spaced grid points specified by the vector \mathbf{x} . Note that this discretization can alter the i th moment by at most $1 - (1 - \epsilon_2)^i$, and hence for an appropriately chosen constant in the assignment of $\epsilon_2 = 1/\exp(O(1/\epsilon))$ and the fact that we care about only the first $\ell = O(1/\epsilon)$ moments, it follows that the sum of the discrepancies due to rounding can be bounded by $1/\exp(O(1/\epsilon))$.

Hence, MomentInverse will return a distribution \mathbf{p}^+ whose moments have discrepancy at most $1/\exp(O(1/\epsilon))$ from the moments of the true spectral distribution, and hence the distance between the recovered distribution and \mathbf{p} is bounded by Proposition 2.3 as $\frac{2\pi}{O(1/\epsilon)} + 3^{O(1/\epsilon)}/\exp(O(1/\epsilon))$. Picking the constant in the big-O notation appropriately yields the theorem.

Finally, note that the computation time is bounded by the collection of the $\exp(O(1/\epsilon)) \log(1/\delta)$ random walks, and solving the linear program on $\exp(O(1/\epsilon))$ variables, which takes time polynomial in the number of variables, and hence is bounded by $\exp(O(1/\epsilon))$, as claimed. \square

While the MomentInverse algorithm returns a distribution \hat{p} described via $\exp(O(1/\epsilon))$ numbers, we note that there is a simple algorithm, computable in $O(n \exp(O(1/\epsilon)))$ time, that will convert \hat{p} into a vector v of length n , with the property that the Wasserstein distance between the spectral distribution p and the distribution associated with v (consisting of n equally-weighted point masses at the locations specified by v) is at most the distance between p and \hat{p} .

DISCRETIZESPECTRUM(n, q):

Input: Distribution q consisting of a finite number of point masses, integer n .

Output: Vector $\mathbf{v} = (v_1, \dots, v_n)$.

- (1) Let $f_q : [0, 1] \rightarrow \mathbb{R}$ be defined to be the non-decreasing function with the property that for X drawn uniformly at random from the interval $[0, 1]$, the distribution of $f_q(X)$ is q .
- (2) Set

$$v_i = \mathbb{E} \left[f_q(X) | X \in \left[\frac{i-1}{n}, \frac{i}{n} \right] \right],$$

and return $\mathbf{v} = (v_1, \dots, v_n)$.

LEMMA 2.5. Consider a distribution p that consists of n equally weighted point masses. Let q be any distribution consisting of a finite number of point masses, and let q_v denote the distribution consisting of n equally weighted point masses located at the values specified by the vector v returned by running Algorithm DiscretizeSpectrum

on inputs n and q . Then the Wasserstein distance between p and q_v satisfies

$$W_1(p, q_v) \leq W_1(p, q).$$

PROOF. Let p_1, \dots, p_n with $p_i \leq p_{i+1}$ denote the support of distribution p . Observe that the earth moving scheme of minimal cost that yields distribution p from distribution q consists of moving the $1/n$ probability mass in distribution q corresponding to the (scaled) conditional distribution $f_q(X)$ conditioned on $X \in [\frac{i-1}{n}, \frac{i}{n}]$ to location p_i . Let q_i denote the i th such conditional distribution. Since, $W_1(p, q_v) = \frac{1}{n} \sum_{i=1}^n |p_i - v_i|$, it suffices to analyze $|p_i - v_i|$ independently for each i . To conclude, note that the contribution of q_i to the Wasserstein distance is simply $\frac{1}{n} \sum_{x \in \text{supp}(q_i)} |x - p_i| \cdot q_i(x) \geq \frac{1}{n} |p_i - \sum_{x \in \text{supp}(q_i)} x \cdot q_i(x)| = \frac{1}{n} |p_i - v_i|$, where for $x \in \text{supp}(q_i)$, we use the shorthand $q_i(x)$ to denote the amount of mass that distribution q_i places on value x . \square

3 APPROXIMATING THE SPECTRUM OF GRAPH LAPLACIANS

In this section we describe how to leverage the results of Section 2.4, namely how to accurately approximate the spectrum of a stochastic matrix, to recover the spectrum of a graph Laplacian, establishing Theorem 1.1. Let $G = (V, E)$, $V = \{1, \dots, n\}$ be an undirected graph and let A be its adjacency matrix. We assume that we have access to data structure that, given an a vertex v , can return a uniformly distributed neighbor of v .

Definition 3.1. The *normalized Laplacian* of a graph G with adjacency matrix A is defined as $L_G = I - D^{-1/2} A D^{-1/2}$, where D is a diagonal matrix whose entries are the vertex degrees.

PROOF OF THEOREM 1.1. Let $M = A D^{-1}$ be the transition matrix of a random walk on G , i.e. $M_{i,j} = \frac{1}{\deg(j)}$ whenever there is an edge between vertex i and j and where $\deg(j)$ denotes the degree of vertex j . Note that $M = D^{1/2} (I - L_G) D^{-1/2}$ and so M is similar to the real valued symmetric matrix $I - L_G$. Thus, M is a stochastic matrix that can be written as $Q \Lambda Q^{-1}$ and the i -th largest eigenvalue λ_i of L_G corresponds to an i -th smallest eigenvalue $1 - \lambda_i$ of M (in particular, the eigenvalues are real).

Hence an approximation $\hat{\lambda} = \hat{\lambda}_1, \dots, \hat{\lambda}_n$ of the spectrum of M with error ϵ can be mapped to an approximation $\hat{\lambda}^G = \hat{\lambda}_1^G, \dots, \hat{\lambda}_n^G$ of the spectrum of L_G that has error ϵ by setting $\hat{\lambda}_i^G = 1 - \hat{\lambda}_i$, yielding Theorem 1.1. \square

4 AN IMPROVED ALGORITHM FOR BOUNDED DEGREE PLANAR GRAPHS

In this section we describe an improved algorithm for bounded degree planar graphs and, more generally, minor-closed bounded-degree graphs, establishing Theorem 1.2. We need two main tools to obtain this result. The first one is a lemma that shows that the Wasserstein distance between the spectra of graphs is at most twice the graph edit distance.

LEMMA 4.1. Let $G = (V, E)$ and $H = (V, E')$ be two graphs. Then

$$|V| \cdot W_1(\lambda, \lambda') \leq 2G\Delta H,$$

where $G\Delta H$ denotes the number of edges that need to be changed to transform G into an isomorphic copy of H and λ and λ' are the spectra of G and H , respectively.

PROOF. We first recall the variational characterization of eigenvalues for a symmetric $n \times n$ matrix Q :

$$\lambda_k(Q) = \min\{\lambda \mid \exists k - \text{subspace } F \subset \mathbb{R}^n \text{ s.t. } \forall x \in F \ x^t Q x \leq \lambda x^t x\}$$

Let $U \subset \mathbb{R}^V$ be the subspace of functions that vanish on the vertices incident to at least an edge that is in one of the graphs G and H only. By assumption, the codimension of U is at most $G\Delta H$. Now, it is easy to see that the (normalized) Laplacian quadratic forms L_G and L_H coincide on U . For $0 \leq \lambda \leq 2$, let $f_G(\lambda)$ (resp. $f_H(\lambda)$) be the fraction of eigenvalues of G (resp. H) that are below λ . From the variational principle, for a given λ , there is a $f_G(\lambda)|V|$ -subspace $W \subset \mathbb{R}^V$ such that $\forall x \in W \ x^t L_G x \leq \lambda x^t x$. The subspace $W \cap U$ is at least $f_G(\lambda)|V| - G\Delta H$ dimensional and because the two quadratic forms coincide on it, it witnesses that $f_H(\lambda)|V| \geq f_G(\lambda)|V| - G\Delta H$ using the variational principle. By symmetry, $|f_G - f_H| \leq G\Delta H/|V|$.

Since f_G and f_H coincide outside $[0, 2]$, we see that $\int |f_G - f_H| \leq 2G\Delta H/|V|$. The latter integral is the area between the graphs of f_H and f_G . Now, switching axes, these graphs become the graphs of the inverse cumulative distribution functions of the spectral measures of G and H . Since the Wasserstein distance is the L_1 distance between (inverse) cumulative distribution functions, the result follows. \square

The second tool is an algorithmic gadget called a “planar partitioning oracle”. It is well known that by applying the planar separator theorem [20] multiple times one can partition a planar graph with maximum degree d into connected components of size $O(d/\epsilon^2)$ by removing ϵn edges from the graph. A planar partitioning oracle provides local access to such a partition.

Definition 4.2 ([8]). We say that \mathcal{O} is an (ϵ, k) -partitioning oracle for a class \mathcal{C} of graphs if given query access to a graph $G = (V, E)$ in the adjacency-list model, it provides query access to a partition P of V . For a query about $v \in V$, \mathcal{O} returns $P[v]$. The partition has the following properties:

- P is a function of the graph and random bits of the oracle. In particular, it does not depend on the order of queries to \mathcal{O} .
- For every $v \in V$, $|P[v]| \leq k$ and $P[v]$ induces a connected graph in G .
- If G belongs to \mathcal{C} , then $|\{(v, w) \in E : P[v] \neq P[w]\}| \leq \epsilon n$ with probability $9/10$.

We will leverage a partitioning oracle by Levi and Ron:

THEOREM 4.3 ([18]). *For any fixed graph H there exists an $(\epsilon, O(d^2/\epsilon^2))$ -partition-oracle for H -minor free graphs that makes $(d/\epsilon)^{O(\log(1/\epsilon))}$ queries to the graph for each query to the oracle. The total time complexity of a sequence of q queries to the oracle is $q \log q \cdot (d/\epsilon)^{O(\log(1/\epsilon))}$.*

The partitioning oracle provides us access to a partition of a minor-closed graph $G = (V, E)$ into small connected components. This partition is obtained by removing at most ϵn edges. Let us call the graph that consists of these connected components $H(V, E')$. By our first lemma the spectra of G and H have Wasserstein distance at most ϵ . This means that if we can approximate the spectrum of a graph with small connected components, then we can also estimate

the spectrum of a minor-closed bounded degree graph using the partitioning oracle from above.

We now provide a simple algorithm that samples eigenvalues from the spectrum of a graph with small connected components.

SMALLCCSPECTRUM($H = (V, E)$):

Input: Graph $H = (V, E)$ with small connected components.

Output: A random eigenvalue of the normalized Laplacian of H .

- (1) Sample a vertex $v \in V$ uniformly at random
- (2) Compute the connected component $C(v)$ of v
- (3) Return a random eigenvalue of the normalized Laplacian of $C(v)$

LEMMA 4.4. *Algorithm SmallCCSpectrum samples a random eigenvalue from H . If all connected components are of size at most t then the running time of the algorithm is $O(t^3)$.*

PROOF. First we observe that the spectrum of H is the union of the spectrum of its connected components. Indeed, given an eigenvalue with corresponding eigenvector of a connected component $C(v)$ of H we observe that extending the eigenvector with 0 will yield an eigenvector of H with the same eigenvalue.

Next we observe that the algorithm returns a uniformly distributed eigenvalue of H . Let us fix an eigenvalue λ_i belonging to connected component C . The probability to sample λ_i is the probability to sample a vertex from the connected component (which is $|V(C)|/|V|$) times the probability that λ_i is sampled from the $|V(C)|$ eigenvalues of the connected component, which is $1/|V(C)|$. Hence the probability to sample λ_i is $1/|V|$. \square

Theorem 1.2. *Let \mathcal{G} be a family of graphs of maximum degree at most d that does not contain a forbidden minor H . Then one can approximate the spectrum of G in Wasserstein distance up to an additive error of ϵ in time $(\frac{d}{\epsilon})^{O(\log(1/\epsilon))}$.*

PROOF. The approximation guarantee follows from the relation between edit distance and Wasserstein distance and when we estimate the spectrum using polynomially (in $1/\epsilon$) many calls to algorithm SmallCCSpectrum. The running time then follows from the running time of the planar partitioning oracle (where the additional factors in $1/\epsilon$ are absorbed by the O -notation in the exponent). \square

5 EXPERIMENTS

In this section we demonstrate the practical viability of our spectrum estimation approach on both real-world and synthetically generated graph. As our approach is the first proposed algorithm for estimating the spectrum that runs in time sublinear in the size of the graph (and actually runs in time independent of the size of the graph), there is no natural performance comparison to run. Instead, we first demonstrate that the performance in practice scales as a function of the number of random walk used to estimate the spectral moments, and not as a function of the graph size. We then investigate the spectra of 15 undirected network datasets that are publicly available on the Stanford Large Network Dataset Collection [17].

All experiments were run in Matlab on a MacBook Pro laptop. For each network, we ran our spectrum estimation algorithm 20 times and averaged the 20 returned spectra. Each of the spectra was obtained by simulating s independent random walks of length 20 steps each, and then leveraging the APPROXSPECTRALMOMENT algorithm of Section 2.2 to estimate the first 20 spectral moments. The synthetic experiments illustrate the effect of varying the number of walks, s . For the real-world networks, the number of random walks was set to $s = 10,000$. The moment estimates were then provided as input to the MOMENTINVERSE algorithm, which returned an approximation to the spectrum supported on a mesh of $t = 1,000$ equally spaced points in the interval $[0, 2]$; hence the linear program of the MomentInverse algorithm has 1,000 variables and could be solved easily. The reason for repeating the spectrum approximation algorithm several times and averaging the returned spectra was due to the tendency of the linear program to output sparsely supported spectra—perhaps due to the particular instabilities of Matlab’s linear program solver. Empirically, averaging several of these runs seemed to yield a very consistent spectrum (that agreed closely with the ground truth for those networks on which we could compute the exact spectrum).

5.1 Synthetic Experiments

We evaluated our approach on synthetic graphs generated according to the Lancichinetti-Fortunato-Radicchi (LFR) random network generator, provided by [13]. We considered graphs of a variety of sizes from 100 nodes to 20,000 nodes, and with average degree either 15 or 30. For each parameter setting, we generated an instance of the random graph on which we ran our recovery algorithm, and varied the number of random walks from 20 to 2 million. For each setting, we evaluated the average Wasserstein recovery error (with respect to the actual spectrum), averaged over 20 independent trials.

Figure 1 depicts the result of these tests, showing that 1) the average recovery error is essentially independent of the graph size, and depends only on the number of walks, and 2) the error scales roughly as the inverse of the logarithm of the number of walks, which is consistent with the theory suggested by Theorem 1.1, which shows that $\exp(O(1/\epsilon))$ walks are sufficient to obtain error ϵ .

5.2 Experiments on Real-World Networks

We also applied our approach to 15 publicly available real-world networks. These datasets include three road networks (ranging from 1M nodes to 1.9M nodes), six co-authorship networks including the DBLP collaboration network (317k nodes, 1M edges), and six social networks including small portions of Facebook (4k nodes, 88k edges), Twitter (81k nodes, 1.7M edges), and Google+ (107k nodes, 13M edges), as well as the LiveJournal social graph, (4M nodes, 34M edges), Orkut (3M nodes, 117M edges), and a portion of the Youtube user follower graph (1M nodes, 2.9M edges).

For the smaller networks—those with $< 50k$ nodes, we computed the exact spectrum in addition to running our spectrum estimation algorithm. In all cases, our reconstruction achieved an earthmover distance at most 0.03 from the actual spectrum. For the larger networks, it was computationally infeasible to compute the exact spectrum.

Discussion of Network Spectra: The recovered spectra of the fifteen graphs considered are depicted in Figure 2. Each plot depicts a representation of the vector of eigenvalues, sorted from largest to smallest. The true spectrum (red points) consisting of $|V|$ eigenvalues is depicted on the same axes as the approximated spectrum (blue points) corresponding to the sorted list of recovered 1,000-quantiles. (If the recovered quantiles had no error, then the two plots would be identical, up to the discrepancy in granularity which is not apparent at this resolution.)

The emphasis of this work is the proposal of an efficient algorithm for recovering the spectrum, as opposed to a detailed analysis of the structural implications of the observed spectra of the graphs considered. Nevertheless, the spectra exhibit several striking phenomena worth discussing.

The most immediate observations are that the spectra of the different classes of network look quite distinct, with the road networks exhibiting very distinctive linear spectra, the collaboration networks looking fairly similar to each other, and the social networks looking anti-symmetric about their midpoints and looking more varied. We now discuss some of the structural insights into these networks that can be gleaned from the spectra.

In hindsight, the linearity of the spectra of the road networks should not be unexpected. Many portions of road networks resemble rectangular grids, and, for a random walk on such a grid, the probability of returning to the origin after t timesteps will scale as $1/t$ for even t (and will be 0 for odd t). These return probabilities correspond to the moments of a uniform distribution supported on the interval $[-1, 1]$, which is then translated to the uniform distribution over $[0, 2]$ when the spectrum of the Laplacian is obtained from that of the corresponding stochastic matrix. Hence, for these road networks, the linear spectrum implies that, locally, the networks tends to behave like rectangular grids. Keep in mind that this sort of insight into these road networks could not have been readily gleaned from the first several eigenvalues, or other standard graph parameters.

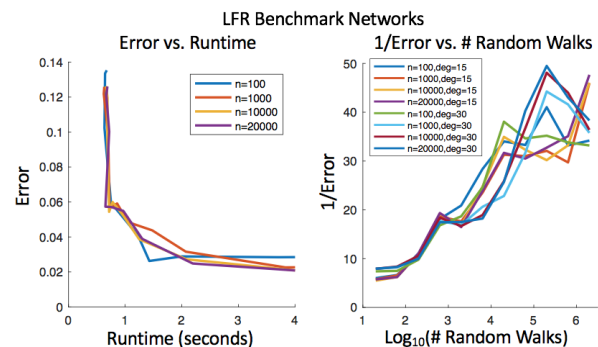


Figure 1: Plots showing the error (left) and $1/\text{error}$ (right) of the spectrum recovered by our algorithm, versus the runtime (left) and $\log(\# \text{ rand walks})$ used to estimate the spectral moments (right). Each line represents the results for one graph, with size ranging from $n = 100$ to $n = 20,000$ nodes. Both plots demonstrate that the performance is independent of the graph size, as claimed. The plot on the right, depicting the inverse of the error, is consistent with Theorem 1.1 suggesting that the error scales as $1/\log(\# \text{ walks})$.

About half of the real-world spectra are close to being anti-symmetric about the middle eigenvalue (particularly the road networks and the social networks—but not the collaboration networks which all exhibit a downward curvature at the middle eigenvalue rather than an inflexion point). This symmetry implies that the odd-moments of the corresponding stochastic matrix are very close to near to zero; namely that odd-length random walks tend to not return to their origin. In the case of the road networks, this makes sense as the networks are essentially locally bi-partite. [The road networks have large grid-like portions.] For other highly symmetric spectra, such as the LiveJournal, Google+, and Orkut, it is likely that almost all of the walks that do return to their origin, do so because they retrace their steps (and hence require an even length path to return), implying that much of the graph behaves locally like a tree or expander.

In contrast to the anti-symmetric spectra of some of the social networks, the collaboration networks have significant downward curvature about the middle eigenvalue. This implies that the networks have significant low-order odd moments (i.e. short random walks of odd length do return to their origin). This implies that there are likely to be many low degree nodes involved in small dense clusters, which give rise to significant return probabilities of length 3 or 5 walks returning. (This, of course, makes perfect sense for collaboration networks.)

One final point about the collaboration networks is that they all have rather similar spectra, despite the DBLP network having a factor of 15 more nodes and edges than some of the other collaboration graphs. This nicely illustrates the phenomena that certain classes of graph have spectra that approach a limiting shape, independent of their size.

The spectra of the social networks appear more diverse. One notable feature—particularly of the Google+, Orkut, and YouTube graphs is the significant number of eigenvalues that are extremely close to 1. These eigenvalues correspond to eigenvectors near the kernel of the adjacency matrix, and hence indicate that these adjacency matrices are close to being significantly rank deficient. In contrast, Facebook, Twitter, and LiveJournal do not have the abundance of eigenvalues very near 1, perhaps suggesting that the nodes tend to be more unique (i.e. everyone has some friends that are not expected).

The spectrum of the Youtube graph is the most curious, and seems to be consistent with the presence of two types of nodes: those for whom random walks return as if they were part of a regular grid, and nodes who contribute to the $\approx |V|/2$ eigenvalues equal to 1 for whom random walks almost never return to their origin.

REFERENCES

- [1] Zhidong Bai, Jiaqi Chen, and Jianfeng Yao. 2010. On estimation of the population spectral distribution from a high-dimensional sample covariance matrix. *Australian & New Zealand Journal of Statistics* 52, 4 (2010), 423–437.
- [2] Fan Chung, Linyuan Lu, and Van Vu. 2003. Spectra of random graphs with given expected degrees. *Proceedings of the National Academy of Sciences* 100, 11 (2003), 6313–6318.
- [3] Fan RK Chung. 1997. *Spectral graph theory*. Vol. 92. American Mathematical Soc.
- [4] David L Donoho, Matan Gavish, and Iain M Johnstone. 2013. Optimal shrinkage of eigenvalues in the spiked covariance model. *arXiv preprint arXiv:1311.0851* (2013).
- [5] László Erdős, Antti Knowles, Horng-Tzer Yau, Jun Yin, et al. 2013. Spectral statistics of Erdős–Rényi graphs I: local semicircle law. *The Annals of Probability* 41, 3B (2013), 2279–2375.
- [6] Illes J Farkas, Imre Derényi, Albert-László Barabási, and Tamas Vicsek. 2001. Spectra of “real-world” graphs: Beyond the semicircle law. *Physical Review E* 64, 2 (2001), 026704.
- [7] Alan Frieze, Ravi Kannan, and Santosh Vempala. 2004. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM (JACM)* 51, 6 (2004), 1025–1041.
- [8] Avinatan Hassidim, Jonathan A Kelner, Huy N Nguyen, and Krzysztof Onak. 2009. Local graph partitions for approximation and testing. In *50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 22–31.
- [9] Nouredine El Karoui. 2008. Spectrum estimation for large dimensional covariance matrices using random matrix theory. *The Annals of Statistics* (2008), 2757–2790.
- [10] Weihao Kong and Gregory Valiant. 2017. Spectrum estimation from samples. *The Annals of Statistics* 45, 5 (2017), 2218–2247.
- [11] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. 2009. On sampling-based approximate spectral decomposition. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 553–560.
- [12] Tsz Chiu Kwok, Lap Chi Lau, Yin Tat Lee, Shayan Oveis Gharan, and Luca Trevisan. 2013. Improved Cheeger’s inequality: analysis of spectral partitioning algorithms through higher order spectral gap. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*. ACM, 11–20.
- [13] Andrea Lancichinetti and Santo Fortunato. 2009. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E* 80, 1 (2009), 016118.
- [14] Olivier Ledoit and Michael Wolf. 2012. Nonlinear shrinkage estimation of large-dimensional covariance matrices. *Annals of Statistics* 40, 2 (2012), 1024–1060.
- [15] Olivier Ledoit and Michael Wolf. 2013. Spectrum estimation: A unified framework for covariance matrix estimation and PCA in large dimensions. *Available at SSRN 2198287* (2013).
- [16] James R Lee, Shayan Oveis Gharan, and Luca Trevisan. 2014. Multiway spectral partitioning and higher-order cheeger inequalities. *Journal of the ACM (JACM)* 61, 6 (2014), 37.
- [17] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. <http://snap.stanford.edu/data>.
- [18] Reut Levi and Dana Ron. 2015. A quasi-polynomial time partition oracle for graphs with an excluded minor. *ACM Transactions on Algorithms (TALG)* 11, 3 (2015), 24.
- [19] Weiming Li and Jianfeng Yao. 2014. A local moment estimator of the spectrum of a large dimensional covariance matrix. *Statistica Sinica* 24 (2014), 919–936.
- [20] Richard J Lipton and Robert Endre Tarjan. 1979. A separator theorem for planar graphs. *SIAM J. Appl. Math.* 36, 2 (1979), 177–189.
- [21] Anand Louis, Prasad Raghavendra, Prasad Tetali, and Santosh Vempala. 2012. Many sparse cuts via higher eigenvalues. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*. ACM, 1131–1140.
- [22] Frank McSherry. 2001. Spectral partitioning of random graphs. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*. IEEE, 529–537.
- [23] Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence* 22, 8 (2000), 888–905.
- [24] Zhao Song, David Woodruff, and Huan Zhang. 2016. Sublinear time orthogonal tensor decomposition. In *Advances in Neural Information Processing Systems*. 793–801.
- [25] Kevin Tian, Weihao Kong, and Gregory Valiant. 2017. Learning Populations of Parameters. In *Neural Information Processing Systems (NIPS)*.
- [26] Christopher KI Williams and Matthias Seeger. 2001. Using the Nyström method to speed up kernel machines. In *Advances in neural information processing systems*. 682–688.
- [27] David P Woodruff et al. 2014. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science* 10, 1–2 (2014), 1–157.

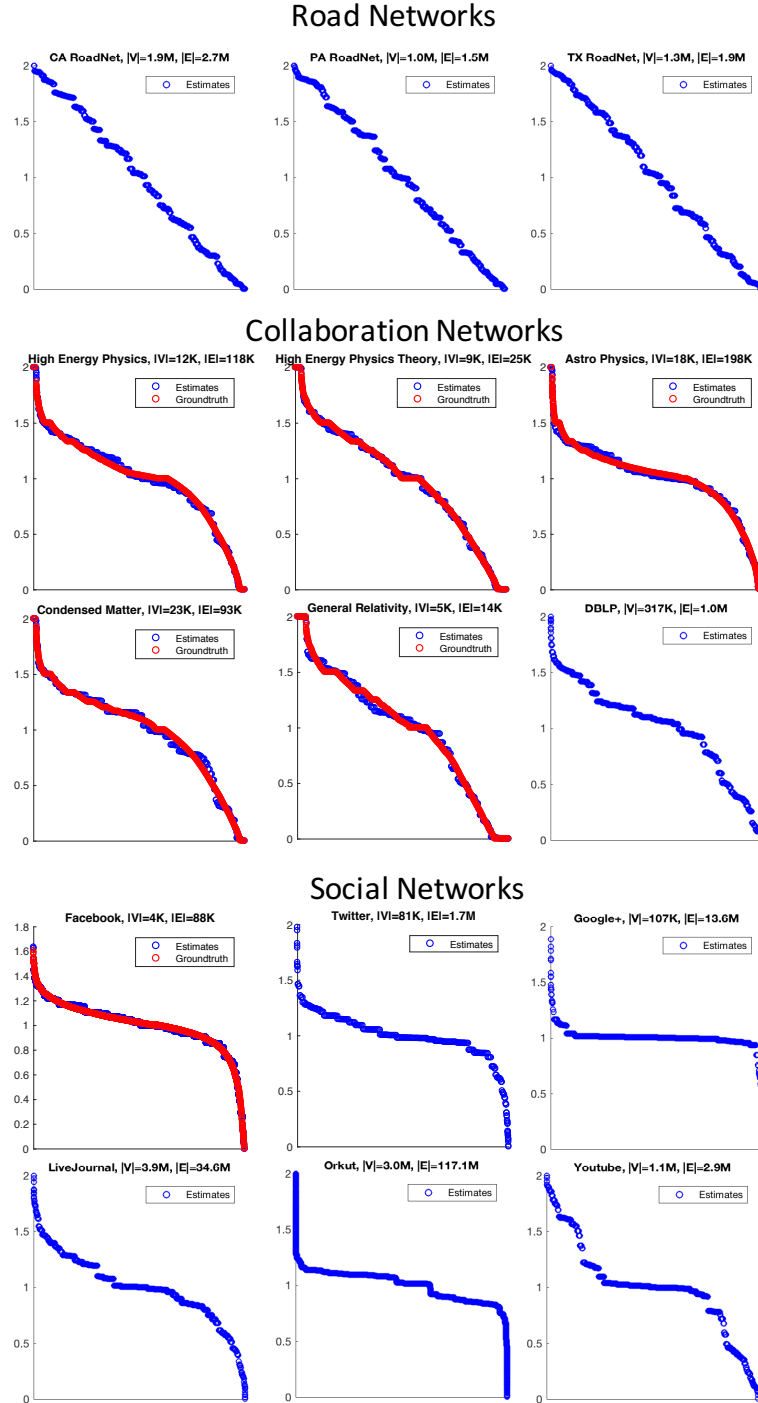


Figure 2: Plots of the results of running our spectrum estimation algorithm on 15 graphs that are publicly available from the Stanford Large Network Dataset Collection. Each plot depicts the sorted vector of eigenvalues (in decreasing order) of the scaled Laplacian of the graph. Viewing the x-axis of each plot as ranging from 0 to 1, for the plot corresponding to a graph $G = (V, E)$, a point at location (x, y) corresponds to saying that the $(x \cdot |V|)$ th largest eigenvalue is y . For the graphs with $< 50k$ nodes, the true spectrum (red) is superimposed on the estimated spectrum (blue). All experiments were run in Matlab on a MacBook Pro laptop, and the estimated spectra required less than 5 minutes of computation time per graph. Matlab code is publicly available from our websites.