Entropic Spectral Learning in Large Scale Networks

Diego Granziol * Oxford University
Binxin Ru * Oxford University
Stefan Zohren Oxford University
Xiaowen Dong Oxford University
Michael Osborne Oxford University
Stephen Roberts Oxford University

Abstract

We present a novel algorithm for learning the spectral density of large scale networks using stochastic trace estimation and the method of maximum entropy. The complexity of the algorithm is linear in the number of non-zero elements of the matrix, offering a computational advantage over other algorithms. We apply our algorithm to the problem of community detection in large networks. We show state-of-the-art performance on both synthetic and real datasets.

1 INTRODUCTION

1.1 The Importance of Networks

Many systems of interest can be naturally characterised by complex networks; examples include social networks (Flake et al., 2000, Leskovec et al., 2007a, Mislove et al., 2007b), biological networks (Palla et al., 2005) and technological networks. The biological cell can be compactly described as a complex network of chemical reactions; trends, opinions and ideologies spread on a social network, in which people are nodes and edges represent relationships; the world wide web is a complex network of documents (web pages representing nodes) with hyper-links denoting edges. A variety of complex graphs have been studied, from scientific collaborations, ecological/cellular networks, to sexual contacts (Albert and Barabási, 2002). For a comprehensive introduction, we recommend the work by Newman (2010).

1.2 Communities and their Importance

One of the most important research questions in network analysis is community detection (Fortunato, 2010). In protein-protein interaction networks, communities are likely to group proteins having the same cellular function (Chen and Yuan, 2006). In the world wide web, communities may correspond to pages dealing with related topics (Dourisboure et al., 2007). In social networks, they may correspond to families, friendship circles, towns and nations.

Communities also have concrete practical applications. For example, clustering geographically close web users with similar interests, could improve web performance by serving them with a dedicated mirror server (Krishnamurthy and Wang, 2000). Identifying clusters of customers with similar purchasing interests allows for the creation of efficient recommender systems (Reddy et al., 2002). For a full review of the importance of clustering and various methods in the literature, we recommend the work by Fortunato (2010).

2 MOTIVATING EXAMPLE

In the fields of statistics, computer science and machine learning, spectral clustering (Ng et al., 2002, Von Luxburg, 2007) has become an incredibly powerful tool for grouping data, regularly outperforming or enhancing other classical algorithms, such as k-means or single linkage clustering.

For most clustering algorithms, including spectral clustering, estimating the number of clusters is a challenging problem (Von Luxburg, 2007), with likelihood, ad-hoc, information theoretic, stability and spectral approaches advocated. In the latter, one analyses the spectral gap in the eigenvalue spectrum, which we refer to as *eigengap* for short. Applying this approach in the era of big-data (where social networks such as Facebook are approaching $n = 2$ billion users) means that standard approaches, such as the canonical Cholesky decomposition, with computational complexity $\mathcal{O}(n^3)$ and storage $\mathcal{O}(n^2)$ are completely prohibitive.

*These two authors contributed equally

We propose a novel maximum entropy algorithm, which we use along with Chebyshev stochastic trace estimation to learn the spectral density of a network. This entails computational complexity $\mathcal{O}(n_{\text{nz}})$, where n_{nz} represents the number of non-zeros elements of the matrix. For a network such as Facebook, where the average user has 300 friends, this potentially represents a speedup of up to $\mathcal{O}(10^{16})$.

We prove a bound on the positive deviation from zero using matrix perturbation theory and the Cauchy-Schwarz inequality for weakly connected clusters and demonstrate its effectiveness on synthetic examples. Having learned the spectrum, we search for a spectral minimum near the origin, corresponding to the *eigengap* and determine the number of clusters. We test our algorithm on both synthetic and real data with available ground truth and show superior performance to the state-of-the-art iterative method, the Lancsoz algorithm.

2.1 Related Work

Krylov subspace methods, using matrix vector products, such as the Lanczos algorithm have been applied to estimating eigengaps and detecting communities with encouraging results (Kang et al., 2011, Ubaru et al., 2017). The computational complexity of the Lanczos algorithm is $\mathcal{O}(n_{\text{nz}} \times m + nm^2) \times d$, where n is the rank of the square matrix $M \in \mathbb{R}^{n\times n}$, d the number of random starting vectors used and m the number of Lanczos steps taken. The computational complexity of our Entropic Spectral Learning is $\mathcal{O}(n_{\text{nz}} \times m) \times d$, where m is the number of moments used, this is a lower computational complexity than Lacnsoz, as there is no need to orthogonalize and store the vectors at each step. The second Lanczos term dominates at $m > n_{\text{nz}}/n$. For many networks in the Stanford Large Network Dataset Collection (SNAP) (Leskovec and Krevl, 2014), such as Amazon, YouTube, Wikipedia and LiveJournal, this condition is reached for low values of m: respectivley, $3, 3, 14, 6$. We find empirically that for good spectral resolution $m_s > 50$, the extra computational overhead for Lanczos is substantial, often over an order of magnitude. We compare our method against Lanczos algorithm on both synthetic and real datasets and our method shows superior performance, as shown in section 7.

3 GRAPH NOTATION

Graphs are the mathematical structure underpinning the formulation of networks. Let $G = (V, E)$ be an undirected graph with vertex set $V = \{v_1, ..., v_N\}$. Each edge between two vertices v_i and v_j carries a non-negative weight $w_{ij} > 0$ and $w_{ij} = 0$ corresponds to two disconnected nodes. For un-weighted graphs we set $w_{ij} = 1$ for two connected nodes. The *adjacency matrix* is defined as $W = (w_{ij})$ with $i, j = 1, ..., n$. The degree of a vertex $v_i \in V$ is defined as

$$d_i = \sum_{j=1}^{n} w_{ij}. \tag{1}$$

The *degree matrix* D is defined as a diagonal matrix that contains the degrees of the vertices along diagonal, i.e., $D_{ii} = d_i$ and zero otherwise. The *unnormalised graph Laplacian matrix* is defined as

$$L = D - W. \tag{2}$$

As G is undirected, i.e., $w_{ij} = w_{ji}$, the adjacency, degree and unnormalised Laplacian matrices are all symmetric. This ensures that the eigenvalues of the Laplacian are real. Another common variant of the Laplacian matrix is the so-called *normalised graph Laplacian* (Chung, 1997)

$$\begin{aligned} \tilde{L} &= D^{-1/2} L D^{-1/2} \\ &= I - \tilde{W} = I - D^{-1/2} W D^{-1/2}, \end{aligned} \tag{3}$$

where $\tilde{W}$ is known as the normalised adjacency matrix[1]. For our analysis we will be using the Laplacian matrix.

Notice that for regular graphs where all the vertices have the same degree, there is no commonly adopted convention on which Laplacian to use as all Laplacians are similar to each other (Von Luxburg, 2007), but outside of this regime the different variants of the Laplacians may vary considerably. For our experiments we use the normalised Laplacian, alternatively we could have used the unnormalised Laplacian and divided by the Gershgorin bound (Gershgorin, 1931) or the number of nodes n.

4 Graph Eigenstructure

Isomorphic graphs are co-spectral. This means that any relabelling of node numbers has no effect on their adjacency matrices after a permutation of rows and columns. Spectral techniques have been used extensively to characterise global network structure (Newman, 2006b) and in practical applications thereof, such as facial recognition/computer vision (Belkin and Niyogi, 2003) and to learn dynamical thresholds (McGraw and Menzinger, 2008). Whilst there exist non-isomorphic cospectral graphs, such as the Saltire pair, computer simulations show that beyond $n \geq 9$ vertices, the fraction f of graphs with co-spectral adjacency and Laplacian matrices decreases and it is conjectured that for $n \to \infty$ that

[1] Strictly speaking, the second equality only holds for graphs without isolated vertices.

$f \to 0$. Furthermore the spectrum of the adjacency and the Laplacian matrices can be used to deduce important quantities, such as the number of vertices and edges, where the graph is regular (fixed girth) or bipartite, the number of closed walks, the number of components and the number of spanning trees (Van Dam and Haemers, 2003).

5 CLUSTERING USING THE EIGENSPECTRA

We reproduce the result that the multiplicity of zero eigenvalues for the graph G indicates its number of connected components. We then use matrix perturbation theory and the Cauchy-Schwarz inequality to show that by adding a small number of edges between the connected components, these eigenvalues are perturbed by a small positive amount. Hence if this perturbation is small compared to the original spectral gap, we can still determine the number of clusters by integrating the spectral density until the first minimum and then multiplying by the dimension of the Laplacian matrix. The following result is well known (Von Luxburg, 2007).

Proposition 1 *Let G be an undirected graph with non-negative weights. Then the multiplicity k of the eigenvalue 0 of the Laplacian $L \in \mathcal{R}^{n\times n}$ is equal to the number of connected components $A_1, ..., A_k$ in the graph. The eigenspace of the eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, ..., \mathbb{1}_{A_k}$.*

For completeness we outline the proof here. Note that, by the definition of the unnormalised Laplacian, we have $L_{ij} = \mathbb{1}_{i=j} \sum_{k=1}^{n} w_{ik} - w_{ij}$ and hence if we set $u_j = 1$ with u_j being the j-th element of $\mathbf{u}$,=

$$\lambda \times u_i = \sum_{j=1} L_{ij} u_j = \sum_{j=1} \left(-w_{ij} + \mathbb{1}_{i=j} \sum_{k=1} w_{ik} \right) = 0 \tag{4}$$

This proves that the vector $\mathbf{u} = [1, ..., 1]^T$ is an eigenvector with eigenvalue $\lambda = 0$. For k connected components, the matrix L has a block diagonal form

$$L = \begin{bmatrix} L_1 & 0 & \dots & 0 \\ 0 & L_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & L_k \end{bmatrix}$$

As is the case for all block diagonal matrices, the spectrum of L is given by the union of the spectra L_i. From the proceeding we know that every Laplacian L_i has an eigenvalue 0 with multipicity 1, hence L has eigenvalue 0 with multiplicity k and corresponding eigenvectors of L are those of L_i filled with 0 at the positions of the other blocks.

Therefore, to learn the number of disconnected components in an arbitary graph, we simply count the number of 0 eigenvalues. However given that real world networks are rarely completely disconnected, this procedure would be of little practical utility.

We hence consider a looser definition of the word cluster and consider groups of nodes containing far greater intra-group connections than inter-group connections. This conforms to our natural intuition of a group or community.

If the graph is connected, but consists of k subgraphs which are "weakly" linked to each other, the unnormalized Laplacian has one zero eigenvalue and all the other eigenvalues positive. This is easily seen by looking at

$$\mathbf{u}^T L \mathbf{u} = \sum_{i,j=1}^{n} w_{ij}(u_i - u_j)^2 \tag{5}$$

which is positive, as $w_{ij} > 0$ and we have proved that a connected graph G has one 0 eigenvalue, hence all other eigenvalues are positive. For small changes in the Laplacian, we expect from matrix perturbation theory (Bhatia, 2013) that the next $k-1$ smallest eigenvalues will be close to 0.

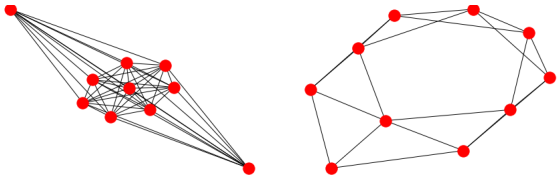
We formalise this intuition by considering a small perturbation of the Laplacian $\tilde{L} = L + \delta L$, where $||\delta L|| \ll ||L||$. It can then be shown that $\forall\, 1 \leq i \leq k$, the difference in the i-th eigenvalue can be written as

$$\lambda_i' - \lambda_i = \delta\lambda_i' = \mathbf{u}_i^T \delta L \mathbf{u}_i \leq ||\delta L||\, ||\mathbf{u}_i^T \mathbf{u}_i|| = ||\delta L|| \tag{6}$$

where we have used the orthonormality of the eigenvectors, the eigenvalues of the unperturbed matrix and the Cauchy-Schwarz inequality.

If we consider the natural variant of the Laplacian, normalised by the number of vertices in the graph, i.e $L_{\text{natural}} = (D - A)/n$, then by adding R vertices between previously disconnected subgraphs, for each vertex, we alter a two diagonal components by $+1$ and two off diagonal components by -1.Thus, our bound goes as R/n by using the Frobenius norm . We note that our derived bound using the Cauchy-Schwarz inequality is exactly the same as Weyl's perturbation theorem for Hermitian matrices, which uses the min-max principle (Bhatia, 2013).

Hence we expect the eigenvalue perturbations to die off as $\mathcal{O}(n^{-1})$ for a constant number of connections between clusters as we increase the number of nodes n in the network. Even if the number of such connections grows with n but is sparse such that the total number is $\mathcal{O}(ns)$



(a) (b)

Figure 1: (a) Erdös-Rényi random graph with $p = 0.1$ and $n = 10$ nodes; (b) Watz-Strogatz with $p = 0.2$, $k = 5$ & $n = 10$

with small sparsity s, the perturbation would only be of order s. For small sparsity s we would expect the spectral gap between the perturbed eigenvalues which were at 0 pre perturbation and the non zero eigenvalues to remain non-negligible. In these cases, we expect our cluster detection algorithm, introduced in the next section to also work.

If we choose to work with the normalised Laplacian defined in (3), then for each new connection between previously disconnected components we get a term of the form

$$\sum_{j=1}\left|\left|\frac{1}{\sqrt{d_i d_j}}-\frac{1}{\sqrt{(d_i+1)d_j}}\right|\right|^2+\frac{2}{(d_i+1)(d_{k+1})}$$
$$+\sum_{l=1}\left|\left|\frac{1}{\sqrt{d_k d_l}}-\frac{1}{\sqrt{(d_k+1)d_l}}\right|\right|^2 \quad (7)$$

where nodes k and i are being connected and nodes j and l are the nodes connected to k and i, respectively. By taking the degrees to be a fraction of the total number of nodes n and taking n to be large we observed a similar n^{-1} scaling. The idea of strong communities being nearly disconnected components, is not novel (McGraw and Menzinger, 2008) and has been used in community detection algorithms (Capocci et al., 2005). However we have not come across a simple exposition of the results from matrix perturbation theory, or the application of the Cauchy-Schwarz inequality to bound the increase in the 0 eigenvalues as a function of node number n or degrees d_i amongst the connected components.

We test the intuition derived by the bound in equation (6) by generating k connected traditional random graphs, shown in Figure 1, of equal size and forming the disjoint union. The number of 0 eigenvalues is given by Proposition 1, which we verify to within numerical precision. We then create a number of links between the clusters and see how the next $k-1$ smallest non-zero eigenvalues change in size. The results for Erdös-Rényi (Erdös and Rényi, 1959) and Watts-Strogatz (Watts and Strogatz, 1998) random graphs of different sizes and parameter values are shown in Tables 1 and 2. We see that for a constant number of connections between the clusters, 1, in this case one interconnected node between the clusters, shown in Figure 2 the smallest non-zero eigenvalues are perturbed from 0 to n^{-1} as expected from our bound. In Table 2, where we create a number of inter-nodal links proportional to the number of nodes, Or alternatively cluster members. With the exception of the $n = 10$ Watts-Strogatz network, we have eigenvalues of similar order. We also test for sizes $n = 500$ and $n = 2000$ and find that the Eigenvalues stay of similar size to the other values in the table.

Table 1: The second and third smallest Laplacian eigenvalues of 3 initially disconnected sets of connected nodes of size n connected by a single inter-node link

n	ERDÖS-RÉNYI ($p = 1$)	WATTS-STROGATZ ($p = 0.3, k = 5$)
10^1	$[8\times10^{-2}, 2\times10^{-1}]$	$[6\times10^{-2}, 2\times10^{-1}]$
10^2	$[9\times10^{-3}, 2\times10^{-2}]$	$[4\times10^{-3}, 1\times10^{-2}]$
10^3	$[9\times10^{-4}, 3\times10^{-3}]$	$[6\times10^{-4}, 1\times10^{-3}]$

Table 2: The second and third smallest Laplacian eigenvalues of 3 initially disconnected sets of connected nodes of size n connected by a number of inter-cluster links $R = 0.1n$ proportional to the number of nodes

n	ERDÖS-RÉNYI ($p = 0.3$)	WATTS-STROGATZ ($p = 0.1, k = 5$)
10^1	$[1\times10^{-1}, 2\times10^{-1}]$	$[2\times10^{-1}, 2\times10^{-1}]$
10^2	$[2\times10^{-1}, 2\times10^{-1}]$	$[2\times10^{-2}, 5\times10^{-2}]$
10^3	$[3\times10^{-1}, 3\times10^{-1}]$	$[2\times10^{-2}, 3\times10^{-2}]$

Hence, the number of communities can be approximated as the number of small eigenvalues close to 0.

As a final remark, note that in the context of the above discussed random graph models, the Laplacian will be a random matrix. There are powerful techniques from random matrix theory which provide analytical expressions for eigenvalue densities of such random matrices (see (Akemann et al., 2011) for an overview).

6 ESTIMATING THE SPECTRAL DENSITY USING MAXIMUM ENTROPY

We now present our novel approach to estimating the spectral density of large scale networks, motivated by the problem of determining the number of clusters therein.

From the previous section, we see that the number of

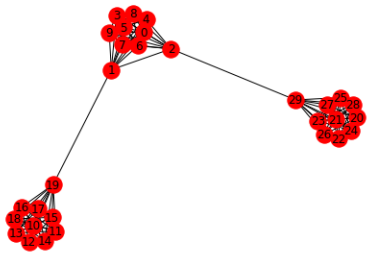


Figure 2: 3 Watts-Strogatz clusters with $p = 0.1, k = 5, n = 10$ where each cluster is connected by a single node.

clusters is equal to the number of near zero eigenvalues. Assuming there is a clear spectral gap, i.e there exists a λ_* which upper bounds the largest perturbed eigenvalue and lower bounds the smallest non-zero eigenvalue pre perturbation, we can write the total number of clusters as

$$C = n \int_0^{\lambda_*} p(\lambda) d\lambda \tag{8}$$

with n being the number of nodes $L \in \mathbb{R}^{n \times n}$ and $p(\lambda)$ denoting the spectral density. A naive eigendecomposition, which would give $p(\lambda)$ as a sum of delta functions, has an infeasible $\mathcal{O}(n^3)$ computational complexity. As previously mentioned in section 2.1, the Lancsoz algorithm, which exploits matrix sparsity by working with matrix vector multiplications, has computational complexity $\mathcal{O}(n_{\text{nz}} \times m + nm^2) \times d$, where for very large sparse matrices, the second term becomes dominant. Given that empirically many social, biological and technical communities are sparse and that all we need for detecting cluster count is an estimation of the eigenvalues and not the eigenvectors, we look for an alternative computationally more effective method of estimating the spectral density. We use the method of maximum entropy, with computational complexity equivalent to Lancsoz without the second term.

6.1 The Method of Maximum Entropy

The method of maximum entropy, hereafter referred to as *MaxEnt* (Pressé et al., 2013) is a procedure for generating the most conservative estimate[2] of a probability distribution possible with the given information, the most non-committal with regard to missing information (Jaynes, 1957).

Intuitively, on a bounded domain, the most conservative distribution, the distribution of maximum entropy, is the one that assigns equal probability to all the accessible states. Hence, the method of maximum entropy can be thought of choosing the flattest, or most equiprobable distribution, satisfying the given constraints.

To determine the spectral density $p(\lambda)$ using MaxEnt, we maximise the entropic functional

$$S = -\int p(\lambda) \log p(\lambda) d\lambda - \sum_i \alpha_i \left[\int p(\lambda) \lambda^i d\lambda - \mu_i \right] \tag{9}$$

with respect to $p(\lambda)$, where $\mathbb{E}[\lambda^i] = \mu_i$ are the power moment constraints on the spectral density, which are estimated using stochastic trace estimation, explained in section 6.2.

The first term in equation (9) is referred to as the Boltzmann-Shannon-Gibbs (BSG) entropy, which has been applied in multiple fields, ranging from condensed matter physics (Giffin et al., 2016) to finance (Buchen and Kelly, 1996, Neri and Schneider, 2012). Recent work in machine learning, has used the method of maximum entropy with stochastic trace estimation to estimate the log determinant of covariance matrices (Fitzsimons et al., 2017), with corresponding analysis on the number of samples and moments required to get a good estimate (Granziol and Roberts, 2017).

Under the axioms of consistency, uniqueness, invariance under coordinate transformations, sub-set and system independence, it can be proved that for constraints in the form of expected values, drawing self-consistent inferences requires maximising the entropy (Pressé et al., 2013, Shore and Johnson, 1980).

Beyond being a computationally cheaper and well theoretically established method for density estimation, we find that the distribution implied by the method of maximum entropy, faithfully represents the true density and well established Lancsoz approximate density, as shown in figure 6. We also note that the bulk of the distribution is better approximated by the MaxEnt method.

6.2 Stochastic Trace Estimation

The intuition behind stochastic trace estimation is that we can accurately approximate the moments of λ with respect to the spectral density $p(\lambda)$ using computationally cheap matrix vector multiplications.

By using the linearity of expectation, trace cyclicity and the standard property of variances, for any random vector v and any matrix A, we can write,

$$\begin{aligned} &\mathbb{E}_v(v^t A v) = \mathbb{E}_v \text{Tr}(v^t A v) = \mathbb{E}_v \text{Tr}(v v^t A) = \text{Tr}\mathbb{E}_v(v v^t A) \\ &= \text{Tr}\left((\mu\mu^t + \Sigma)A\right) = \text{Tr}(\mu^t A \mu) + \text{Tr}(\Sigma A). \end{aligned} \tag{10}$$

Provided that the random vectors have zero mean $\mu =$

[2] With respect to the uniform distribution.

0 and unit variance $\Sigma = I$, equation (10) gives us the required equality in expectation over the set of random vectors

$$\mathbb{E}_v(v^t A v) = \text{Tr}(A). \tag{11}$$

This allows us to generate successive moment estimates of the spectral density $p(\lambda)$ as

$$\text{Tr}(A^m) = \sum_i^n \lambda_i^m = n\mathbb{E}_p(\lambda^m) = \mathbb{E}_v(v^t A^m v). \tag{12}$$

In order to make the computation of the expectation over all random vectors tractable, we replace this expectation with a Monte Carlo average. For d random vectors,

$$\mathbb{E}_v(v^t A^m v) \approx \frac{1}{d}\Big(\sum_{j=1}^{d} v_j^t A^m v_j\Big), \tag{13}$$

where we take the product of the matrix A with the vector v_j, m times, avoiding expensive $\mathcal{O}(n^3)$ matrix-matrix multiplication. We hence calculate the non-central moment expectations in $\mathcal{O}(d \times m \times n_{\text{nz}})$ for sparse matrices, where $d \times m \ll n$. We use these as moment constraints in our MaxEnt formalism to derive the functional form of the spectral density.

The random unit vector v_j can be drawn from any distribution which admits a zero mean and unit variance. Examples include the Gaussian and Rademacher distribution. The latter is proven (Hutchinson, 1990) to give the lowest variance of such estimators. Whilst bounds on the number of samples d required to get within a fractional error ϵ with probability close to one exist (Han et al., 2015), they are not tight enough to be considered practical and many authors have observed that $m \approx 30$ is sufficient for high performance applications in machine learning (Fitzsimons et al., 2017, Ubaru and Saad).

7 EXPERIMENTS

We use $d = 100$ Gaussian random vectors for our stochastic trace estimation, for both MaxEnt and Lanczos (Ubaru et al., 2017). We explain the procedure of going from Adjacency matrix to Laplacian moments in Algorithm 1. When comparing MaxEnt with Lanczos we set the number of moments m equal to the number of Lanczos steps, as they are both matrix vector multiplications in the Krylov subspace. We implement a quadrature MaxEnt algorithm 2. We use a grid size of 10^{-4} over the interval $[0, 1]$ and add diagonal noise on the Hessian to improve conditioning and symmetrise it. We further use Chebyshev polynomial input instead of power moments for improved performance and conditioning. In order to normalise the moment input we use the normalised

Algorithm 1 Learning the Graph Laplacian Moments

Input: Normalized Laplacian $\{\tilde{L}\}$, Number of Probe Vectors d, Number of moments required m
Output: Moments of Normalised Laplacian $\{\mu_i\}$
for i in $1, .., d$ **do**
  Initialise random vector $\vec{z_i} \in R^{1\times n}$
  for j in $1, .., m$ **do**
    $\vec{z_i}' = \tilde{L}\vec{z_i}$
    $\rho_{i,j} = \vec{z_i}^T \vec{z_i}'$
  end for
end for
$\mu_i = 1/d \times \sum_{j=1}^{d} \rho_{i,j}$

Algorithm 2 MaxEnt Algorithm

Input: Moments $\{\mu_i\}$, Tolerance ϵ, Hessian noise η
Output: Coefficients $\{\alpha_i\}$
Initialize $\alpha_i = 0$.
Minimize $\int_0^1 p_\alpha(\lambda)d\lambda + \sum_i \alpha_i \mu_i$
Gradient $\mu_j - \int_0^1 p_\alpha(\lambda)\lambda^j d\lambda$
Hessian $= \int_0^1 p_\alpha(\lambda)\lambda^{j+k} d\lambda$
Hessian $= (H + H')/2 + \eta$
Until $\forall j$ Gradient$_j < \epsilon$

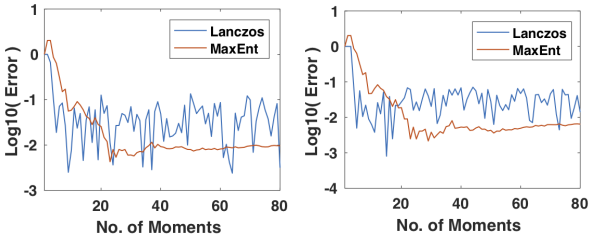
Laplacian with eigenvalues bounded by $[0, 2]$ and divide by 2. We use Python's Scipy implementation of the Newton conjugate gradient algorithm (Jones et al.) for the MaxEnt Lagrange multipliers. We then apply our cluster estimator, Algorithm 3 to both the spectral density derived from our MaxEnt implementation and to that implied by the Lancsoz algorithm. To make a fair comparison we take the output from Lancsoz (Ubaru et al., 2017) and apply kernel smoothing (Lin et al., 2016) before applying our cluster estimator. We explain the details of our kernel smoothing in section 7.2.1.

7.1 Synthetic Data

In order to test the robustness of the approach to networks with clusters of different structures, we implement

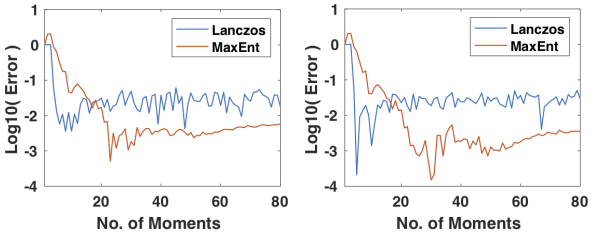
Algorithm 3 Cluster Estimator Algorithm

Input: Lagrange Multipliers α_i, Matrix Dimension n, Tolerance η
Output: Number of Clusters N_c
Initialize $p(\lambda) \to p(\lambda|\alpha_i) = \exp -[1 + \sum_i \alpha_i x^i]$.
Minimize $\lambda*$ s.t $\frac{dp(\lambda)}{d\lambda}|_{\lambda=\lambda*} \leq \eta$
Calculate $N_c = n\int_0^{\lambda*} p(\lambda)d\lambda$



(a) 9 connected Clusters

(b) 30 connected Clusters



(c) 90 connected Clusters

(d) 240 connected Clusters

Figure 3: Log error of community detection using MaxEnt and Lanczos on synthetic networks that contains 9 to 240 clusters

Table 3: Fractional error in community detection for synthetic networks using MaxEnt and Lanczos with 80 moments

# OF CLUSTERS (N)	LANCZOS	MAXENT
9 (270)	$\mathbf{3.20 \times 10^{-3}}$	9.70×10^{-3}
30 (900)	1.41×10^{-2}	$\mathbf{6.40 \times 10^{-3}}$
90 (2700)	1.81×10^{-2}	$\mathbf{5.80 \times 10^{-3}}$
240 (7200)	2.89×10^{-2}	$\mathbf{3.50 \times 10^{-3}}$

a mixture of Erdös-Rényi , Watts-Strogatz and Barabsi-Albert networks using the Python package *NetworkX* and conduct multiple experiments using networks that have from 9 to 240 clusters, with each cluster containing 30 nodes. We connect the nodes between clusters randomly, with a single inter-cluster connection.

Figure 3 shows the community detection errors, expressed in the logarithm to the base 10, for networks of 9, 30, 90, 240 clusters over number of matrix vector calculations (i.e. number of moments). We see that for both methods, the detection error generally decreases as more moments are used. For an equivalent number of matrix vector calculations, MaxEnt outperforms the Lanczos algorithm. As there is no accepted prescription by which we can determine when the spectral minimum has been best learned, the occasional dips in error produced by Lanczos (such as for 15 moments in Figure 3d are unlikely to be replicated in real world experiments.

Table 3 displays the fractional errors in community detection when we apply Lanczos and MaxEnt, both using 80 moments, to synthetic networks of different sizes and cluster numbers. In each case, lower detection error is highlighted in bold. It is evident that MaxEnt outperforms Lanczos as the number of clusters and the network size increase. We observe a general improvement in performance for larger graphs, visible in the differences between fractional errors for MaxEnt and not Lanczos. This is to be expected as the true spectral density

$$p(\lambda) = \frac{1}{n} \sum_{i}^{n} \delta(\lambda - \lambda_i) \tag{14}$$

becomes continuous in the $n \to \infty$ limit and hence we expect the density to be better approximated by a continuous distribution for larger n (Fitzsimons et al., 2017). There are also arguments from the information theoretic literature which state that for macroscopic systems (large n), the distribution of maximum entropy dominates the space of solutions for the given constraints (Caticha, 2000). Essentially this means that for larger and larger systems, assuming that we have incorporated the correct constraints, which for spectral density estimation stochastic trace constraints do (Granziol and Roberts, 2017) the true distribution looks more and more like that of maximum entropy. Furthermore other distributions, i.e a particular realization of Gauss-Lancsoz quaderature (Ubaru et al., 2017) becomes increasingly unlikely.

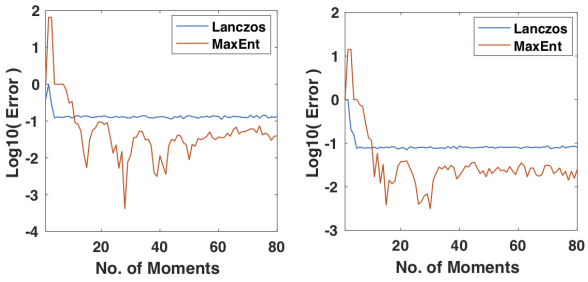
To test the performance of our approach for networks that are too big to apply eigen-decomposition, we also generate two large networks by mixing Erdös-Rényi, Watts-Strogatz and Barabsi-Albert networks. The first large network has a size of 201,600 nodes and comprises 350 interconnected clusters whose size varies from 500 to 1000 nodes. The other large network has a size of 404,420 nodes and comprises interconnected 1355 clusters whose size varies from 200 to 800 nodes. The results in Figure 4 show that our MaxEnt approach outperforms Lanczos for both large synthetic networks.

7.2 Real Data

7.2.1 Small Real World Data

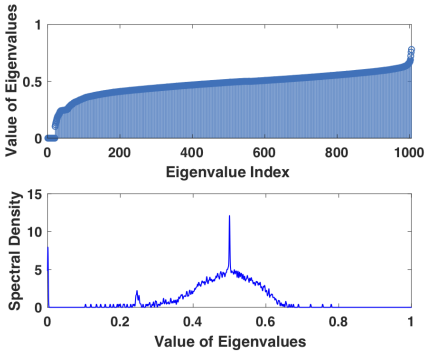
When the number of nodes $n \approx 10^3$, it is possible to compute the eigen-decomposition exactly and hence to benchmark the performance of our algorithm in the real world.

The first real-world dataset we use is the Email network, which is generated using email communication data among $1,005$ members of a large European research institution and is an undirected graph of $n = 1,005$ nodes (Leskovec et al., 2007b). We calculate the ground-truth by computing the eigenvalues explicitly and finding

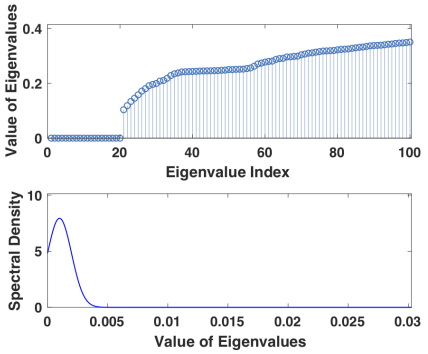


(a) 305 clusters (b) 1,355 clusters

Figure 4: Log error of community detection using MaxEnt and Lanczos on large synthetic networks: a) synthetic network of 201,600 nodes and 305 clusters and b) synthetic network of 404,420 nodes and 1,355 clusters.



(a) Stem Graph and Eigen-Spectrum



(b) Zoomed-in Graphs

Figure 5: Stem graph of the eigen-spectrum of the Email Dataset. The subplot (a) shows all the eigenvalues and the whole eigenvalue spectrum. The subplot (b) is a zoomed-in version of (a), which displays the smallest 100 eigenvalues with a clear spectral gap at 20 and the corresponding spectral density near the origin. The area under the spectral density up to 0.005 multiplied by the number of nodes n predicts the number of clusters.

the spectral gap near 0. As shown in Figure 5, we count 20 very small eigenvalues before a large jump in magnitude and set this as the ground truth for the number of clusters in the network. This corresponds to a drop in spectral eigendensity as displayed in the lower subplot of Figure 5.

We note that this differs from the value of 42 given by the number of departments at the research institute. A likely reason for this ground truth inflation is that certain departments, Astrophysics, Theoretical Physics and Mathematics for example, may collaborate to such an extent that their division in name may not be reflected in terms of node connection structure.

We display the process of spectral learning for both MaxEnt and Lanczos, by plotting the spectral density of both methods against the true eigenvalue spectral density in Figure 6. In order to make a valid comparison, we smooth the implied density using a Gaussian kernel, with $\sigma = 10^{-3}$. We note that both MaxEnt and Lanczos approximate the ground truth better with a greater number of moments/steps m and that Lanczos learns the extrema before the bulk of the distribution.

We plot the log error against the number of moments for both MaxEnt and Lanczos in Figure 7a, with MaxEnt showing superior performance.

We repeat the experiment on the Net Science collaboration network, which represents a co-authorship network of $1,589$ scientists ($n = 1,589$) working on network theory and experiment (Newman, 2006a). The results in Figure 7 show that MaxEnt quickly outperforms the Lanczos algorithm after around 20 moments.

7.2.2 Large Real World Data

For large datasets $n \gg 10^4$, where the Cholesky decomposition becomes completely prohibitive even for powerful machines, we can no longer define a ground truth using a complete eigen-decomposition.

Alternative "ground truths" supplied in (Mislove et al., 2007a), regarding each set of connected components with more than 3 nodes as a community, are not universally accepted. This definition, along with that of self-declared group membership (Yang and Leskovec, 2015), often leads to contradictions with our definition of a community. A notable example being the Orkut dataset, where the number of stated communities is greater than the number of nodes (Leskovec and Krevl, 2014). Beyond being impossible to learn such a value from the eigenspectra, if the main reason to learn about clusters is to partition groups and to summarise networks into smaller substructures, such a definition is undesireable.

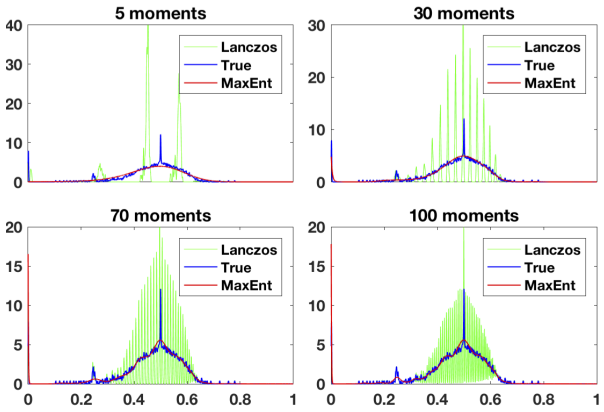


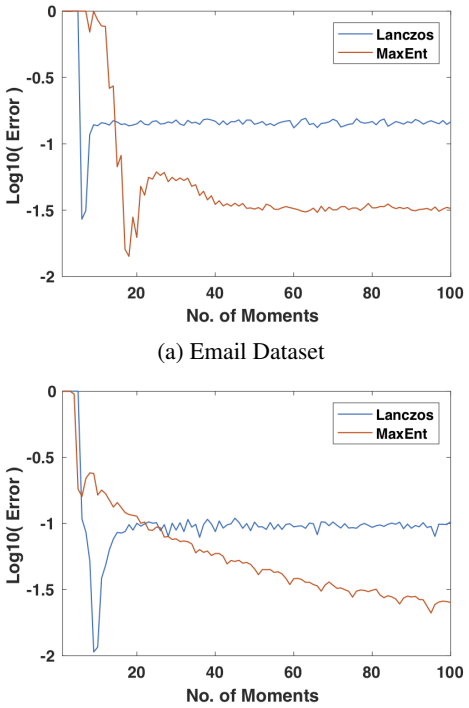
Figure 6: Spectral density for varying number of moments m, for both the MaxEnt and Lanczos algorithm as well as the ground truth.

Table 4: Cluster prediction by MaxEnt for DBLP ($n = 317,080$), Amazon ($n = 334,863$) and YouTube ($n = 1,134,890$).

MOMENTS	40	70	100
DBLP	2.215×10^4	8.468×10^3	8.313×10^3
AMAZON	2.351×10^4	1.146×10^4	1.201×10^4
YOUTUBE	4.023×10^3	1.306×10^4	1.900×10^4

We show that our method continues to faithfully approximate the spectra of large graphs, as shown in Figure 8 by comparing with a kernel smoothed Lanczos approximation. We note that our spectrum displays Gibbs oscillations typical of MaxEnt, which in the case of a badly defined spectral gap (no clear spectral minimum), could lead to spurious minima.

We present our findings for the number of clusters in the DBLP ($n = 317,080$), Amazon ($n = 334,863$) and YouTube ($n = 1,134,890$) networks (Leskovec and Krevl, 2014) in Table 4 for a varying number of moments. We see that for both the DBLP and Amazon networks, the number of clusters N_c seems to converge with increasing moments number m, whereas for YouTube such a trend is not visible. This can be explained by looking at the approximate spectral density of the networks implied by Maximum Entropy in Figure 8. For both DBLP and DBLP, Figures 8a and 8b respectively we see that our method implies a clear spectral gap near the origin, indicating the presence of clusters. Whereas for the YouTube dataset, shown in Figure 8c, no such clear spectral gap is visible and hence the number of clusters cannot be estimated accurately.



(a) Email Dataset

(b) NetScience Dataset

Figure 7: Log error of community detection using MaxEnt and Lanczos algorithms on for differing number of moments m.

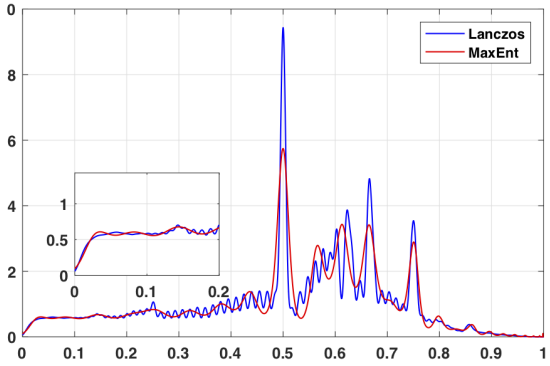
8 CONCLUSION

We present an algorithm for learning the spectral density of large networks and propose a method for using the spectrum to learn the number of clusters within the network. We experimentally validate our approach on both synthetic and real world data.

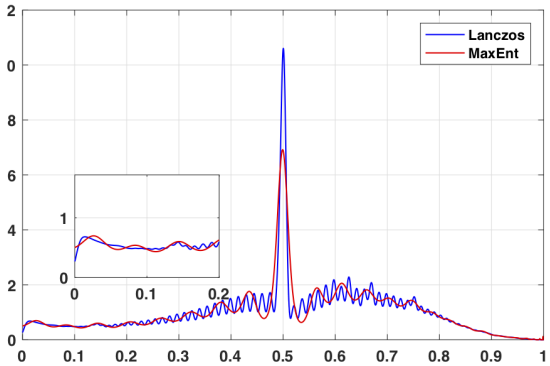
The major advantage of our algorithm using maximum entropy is its computational complexity which is $\mathcal{O}(n_{\text{nz}}md)$, where n_{nz} is the number of non-zeros of the matrix, m is the number of moments we use and d the number of random starting vectors. When compared to state-of-the-art algorithms using Lanczos iteration, our algorithm is seen to have smaller prediction errors compared to the ground truth.

As a byproduct, we present an alternative derivation of a bound on eigenvalue perturbations and relate this to the ratio of inter-cluster to intra-cluster links before, which must be satisfied for our methodology to work well.

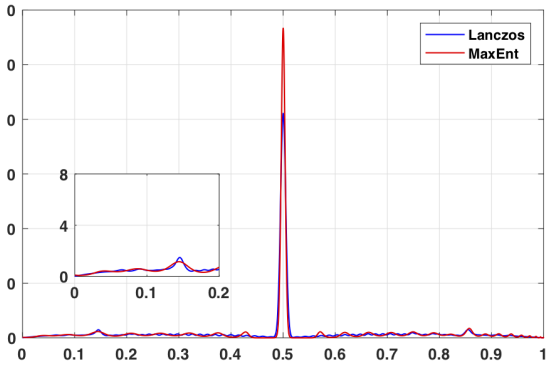
Finally, we also note that compared to Lancsoz our Maximum Entropy approach more faithfully reconstructs the bulk of the spectrum and does not require a smoothing parameter. Future extensions could look into approxi-



(a) DBLP



(b) Amazon



(c) Youtube

Figure 8: Spectral Density for DBLP, Amazon and YouTube Dataset using $m = 100$ by MaxEnt and Lanczos Approximation

mating the divergence between real large networks using our methodology.

References

G. Akemann, J. Baik, and P. D. Francesco. *The Oxford Handbook of Random Matrix Theory*. Oxford University Press, 2011.

R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1): 47, 2002.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

R. Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

P. W. Buchen and M. Kelly. The Maximum Entropy Distribution of an Asset inferred from Option Prices. *Journal of Financial and Quantitative Analysis*, 31 (01):143–159, 1996.

A. Capocci, V. D. Servedio, G. Caldarelli, and F. Colaiori. Detecting communities in large networks. *Physica A: Statistical Mechanics and its Applications*, 352 (2-4):669–676, 2005.

A. Caticha. Maximum entropy, fluctuations and priors. 2000.

J. Chen and B. Yuan. Detecting functional modules in the yeast protein–protein interaction network. *Bioinformatics*, 22(18):2283–2290, 2006.

F. R. Chung. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

Y. Dourisboure, F. Geraci, and M. Pellegrini. Extraction and classification of dense communities in the web. In *Proceedings of the 16th international conference on World Wide Web*, pages 461–470. ACM, 2007.

P. Erdös and A. Rényi. On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

J. Fitzsimons, D. Granziol, K. Cutajar, M. Osborne, M. Filippone, and S. Roberts. Entropic trace estimates for log determinants, 2017.

G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–160. ACM, 2000.

S. Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

S. A. Gershgorin. Über die Abgrenzung der Eigenwerte einer Matrix. *Izvestija Akademii Nauk SSSR, Serija Matematika*, (6):749–754, 1931.

A. Giffin, C. Cafaro, and S. A. Ali. Application of the Maximum Relative Entropy method to the Physics of Ferromagnetic Materials. *Physica A: Statistical Mechanics and its Applications*, 455:11 – 26, 2016. ISSN 0378-4371.

D. Granziol and S. J. Roberts. Entropic determinants of massive matrices. In *2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017*, pages 88–93, 2017.

I. Han, D. Malioutov, and J. Shin. Large-scale log-determinant computation through stochastic Chebyshev expansions. In *International Conference on Machine Learning*, pages 908–917, 2015.

M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 19(2):433–450, 1990.

E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, May 1957.

E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python. URL `http://www.scipy.org/`. [Online; accessed ¡today¿].

U. Kang, B. Meeder, and C. Faloutsos. Spectral analysis for billion-scale graphs: Discoveries and implementation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 13–25. Springer, 2011.

B. Krishnamurthy and J. Wang. On network-aware clustering of web clients. *ACM SIGCOMM Computer Communication Review*, 30(4):97–110, 2000.

J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5, 2007a.

J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007b.

L. Lin, Y. Saad, and C. Yang. Approximating spectral densities of large matrices. *SIAM Review*, 58(1):34–65, 2016.

P. N. McGraw and M. Menzinger. Laplacian spectra as a diagnostic tool for network structure and dynamics. *Physical Review E*, 77(3):031102, 2008.

A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC'07)*, San Diego, CA, October 2007a.

A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42. ACM, 2007b.

C. Neri and L. Schneider. Maximum Entropy Distributions inferred from Option Portfolios on an Asset. *Finance and Stochastics*, 16(2):293–318, 2012. ISSN 1432-1122.

M. Newman. *Networks*. Oxford University Press, 2010.

M. E. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006a.

M. E. Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006b.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *nature*, 435(7043):814, 2005.

S. Pressé, K. Ghosh, J. Lee, and K. A. Dill. Principles of Maximum Entropy and Maximum Caliber in Statistical Physics. *Reviews of Modern Physics*, 85:1115–1141, Jul 2013.

P. K. Reddy, M. Kitsuregawa, P. Sreekanth, and S. S. Rao. A graph based approach to extract a neighborhood customer community for collaborative filtering. In *International Workshop on Databases in Networked Information Systems*, pages 188–200. Springer, 2002.

J. Shore and R. Johnson. Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross-Entropy. *IEEE Transactions on information theory*, 26(1):26–37, 1980.

S. Ubaru and Y. Saad. Applications of trace estimation techniques.

S. Ubaru, J. Chen, and Y. Saad. Fast estimation of tr(f(a)) via stochastic Lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1075–1099, 2017.

E. R. Van Dam and W. H. Haemers. Which graphs are determined by their spectrum? *Linear Algebra and its applications*, 373:241–272, 2003.

U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440, 1998.

J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.