Entropic Spectral Learning in Large Scale Networks

Abstract

We present a novel algorithm for learning the spectral density of large scale networks using stochastic trace estimation and maximum entropy method. The algorithms complexity is linear in the number of non-zeros of the matrix which is a great computational advantage when compared to other algorithms. We apply our algorithm to the problem of community detection in large networks and to determining the similarity between different networks using the Jensen Shannon divergence. We show state-of-the-art performance on both synthetic and real datasets against compareable methods.

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. 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 reccomender 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 (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, estimating the number of clusters is an open 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 2 billion users) means that standard approaches, such as the canonical Cholesky decom-

position, with computational complexity $\mathcal{O}(n^3)$ and storage $\mathcal{O}(n^2)$ are completely prohibitive.

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 $\mathcal{O}(n_{\rm nz})$, where $n_{\rm nz}$ 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[1] 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. We further provide analytical formulae for the network differential entropy and divergence between networks, which have been used to quantify graph stfucture and similarity respectively (Takahashi et al., 2012).

2.1 Related Work

Krylov subspace methods, using matrix vector products, such as the Lanczos algorithm have been applied to estimating eigengaps and to detect communities with encouraging results (Kang et al., 2011, Ubaru et al., 2017). The computational complexity of the Lanczos algorithm is $\mathcal{O}(n_{\rm nz} \times m_s + nm_s^2) \times d$, 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_s the number of Lanczos steps taken. The computational complexity of our Entropic Spectral Learning is $\mathcal{O}(n_{\rm nz} \times m) \times d$, where m is the number of moments used, as there is no need to orthogonalise the Lanczos vectors at each step. The second Lanczos term dominates at $m_s > n_{\rm nz}/n$. 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 $m_s = [3, 3, 14, 6]$ respectively. We find empirically that for good spectral resolution $m_s > 50$, the extra computational overhead for Lanczos is substantial. We compare our method against Lanczos algorithm on both synthetic and real datasets and our method shows superior performance.

[1] Corresponding to the eigengap.

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., $D_{ii} = d_i$. The **unnormalised graph Laplacian matrix** is defined as

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

As G is undirected $w_{ij} = w_{ji}$, which means that the weight matrix is symmetric and hence W is symmetric. Since every diagonal matrix is symmetric, the unnormalised Laplacian matrix is also symmetric. This ensures that the eigenvalues of the Laplacian are real. Another common characterisation of the Laplacian matrix is the **Normalised Laplacian matrix** (Chung, 1997)

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

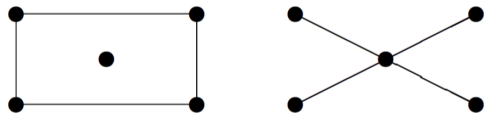
where $\tilde{W}$ is known as the normalised adjacency matrix[2]. For our analysis we will be using the Laplacian matrix. There is no commonly adopted convention on which Laplacian to use. For regular graphs, where most vertices have the same degree, all Laplacians are similar to each other (Von Luxburg, 2007), but outside this regime, they vary considerably. For our experiments, we use the normalised Laplacian but we could have just as well 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

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

facial recognition/computer vision (Belkin and Niyogi, 2003) and to learn dynamical thresholds (McGraw and Menzinger, 2008).



$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Whilst there exist non-isomorphic cospectral graphs, such as the Saltire pair shown above, computer simulations show that beyond $n \geq 9$ vertices, the fraction f of graphs with cospectral adjacency and Laplacian matrices decreases and it is conjectured that for $n \to \infty$ that $f \to 0$. Furthermore the spectrum of the adjacency and the Laplacian matrices can be used to deduce important quantities, such as the number of vertices/edges, where the graph is regular (fixed girth) / 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 well-known result that the number of zero eigenvalues of the graph G indicates the number of connected components. We then use matrix perturbation theory and the Cauchy-Schwarz inequality to show that by adding a small amount 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.

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 $\mathbb{1}_{A_1}, ... \mathbb{1}_{A_k}$*

An intuitive proof can be found in (Von Luxburg, 2007), which we outline for completeness. Note that by the definition of the unnormalised Laplacian, that $L_{ij} = \mathbb{1}_{i=j} \sum_{k=1}^{n} w_{ik} - w_{ij}$ and hence for $u_j = 1$

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

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

$$\begin{bmatrix} L_1 & \dots & \dots \\ \dots & \ddots & \dots \\ \dots & \dots & L_k \end{bmatrix}$$

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.

Hence 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.

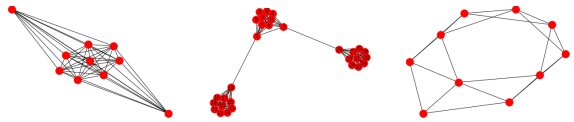
If the graph is connected, but consists of k subgraphs which are "weakly" linked to each other, the unnormalized Laplacian will have one zero eigenvalue and all the other eigenvalues positive. This is easily seen by looking at

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

which is positive semi-definite, 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 to be close to 0. If we consider a small perturbation of the Laplacian $\tilde{L} = L + \delta L$, where $||\delta L|| \ll ||L||$, then it can be shown that $\forall\, 1 \leq i \leq k$, the difference in the i-th eigenvalue can be written as

$$\lambda_i' - \lambda_i = u_i^t \delta L u_i \leq ||\delta L||\, ||u_i^t u_i|| = ||\delta L|| \quad (6)$$

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



(a) (b) (c)

Figure 1: (a) Erdos-Renyi random graph with $p = 0.1$ and $n = 10$ nodes; (b) 3 Watz-Strogatz clusters with $p = 0.1, k = 5, n = 10$ where each cluster is connected by a single node; (c) Watz-Strogatz with $p = 0.2$, $k = 5$ & $n = 10$

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 $\mathcal{O}(ns)$ with small sparsity s, the perturbation would only be of order s and for small sparsity s we would expect our algorithm to also work in those cases.

If we choose to work with the normalised Laplacian (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.

Table 1: Second and Third smallest eigenvalues of 3 initially disconnected sets of connected nodes of size n connected by a single inter-node link

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

Table 2: Second and Third smallest eigenvalues of 3 initially disconnected sets of connected nodes of size n connected by a number of links $R = 0.1n$ proportional to the number of nodes

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

We test the intuition derived by this bound by generating k connected traditional random graphs 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 Erdos-Renyi (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, the smallest non-zero eigenvalues are peturbed 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[3], with the exception of the very small 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.

Hence, the number of communities can be approximated as the number of small eigenvalues close to 0. For a continous spectral density this can be written as

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

with n being the number of nodes and $\lambda*$ being the location of the first spectral minimum corresponding to the

[3] Or alternatively cluster members.

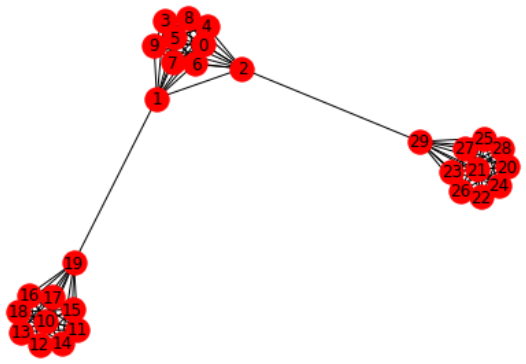


Figure 2: 3 Watts-Strogatz clusters with $p = 0.1, k = 5, n = 10$ where each cluster is connected by a single node.

spectral gap.

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

The method of maximum entropy, hereafter referred to as MaxEnt (Pressé et al., 2013) is a procedure for generating the most conservative estimate[4] of a probability distribution possible with the given information, the most non-committal with regard to missing information (Jaynes, 1957).

To determine the spectral density $p(\lambda)$, 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.

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). 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).

[4]With respect to the uniform distribution.

6.1 Stochastic Trace Estimation

Using the expectation of quadratic forms, for any multivariate random variable v with mean m_z and variance Σ, we can write

$$\mathbb{E}(zz^t) = m_z m_z^t + \Sigma \xrightarrow[m_z=0]{\Sigma=I} I, \tag{10}$$

where in the last relation we have assumed that the variable possesses zero mean and unit variance. By the linearity of trace and expectation for any $m_z \geq 0$

$$\sum_{i=1}^{n}\lambda_i^{m_z} = n\mathbb{E}_\mu(\lambda^{m_z}) = \mathrm{Tr}(IK^{m_z}) = \mathbb{E}(zK^{m_z}z^t). \tag{11}$$

We approximate the expectation over all random vectors with a simple Monte Carlo average. i.e for d random vectors ,

$$\mathbb{E}(zK^{m_z}z^t) \approx \frac{1}{d}\left(\sum_{j=1}^{d} z_j K^{m_z} z_j^t\right), \tag{12}$$

where we take the product of the matrix K with the vector z_j, m times, so as to avoid costly $\mathcal{O}(n^3)$ matrix matrix multiplication. This allows us to calculate the non central moment expectations in $\mathcal{O}(dm \times n_{\mathrm{nz}})$ for sparse matrices, where $d \times m \ll n$.

The random unit vector z_j can be drawn from any distribution, such as a Gaussian. Choosing the components of z_j to be i.i.d Rademacher random variables i.e $P(+1) = P(-1) = \frac{1}{2}$ via Hutchinson's method (Hutchinson, 1990) has the lowest variance of such estimators. Loose bounds exist on the number of samples d required to get within a fractional error ϵ with probability close to one (Han et al., 2015).

6.2 Analytical forms for the Differential Entropy and Divergence from MaxEnt

In other work using either the exact eigen-decomposition (Takahashi et al., 2012) or Chebyshev/Lanczos (Ubaru and Saad) as the differential entropy of delta function

$$\mathcal{S}(\delta(x)) = 1/2\log(2\pi e\sigma^2) \xrightarrow{\sigma\to 0} -\infty. \tag{13}$$

Gaussian Kernel regression, along with a prescriptions for the Bandwidth using the Sturges criterion is used.

Given that neither the differential entropy of a Gaussian mixture nor the Kullback-Leiber/Shannon-Jensen divergence between a Gaussian mixture has an analytical form (Hershey and Olsen, 2007), this too must be approximated using either Monte-Carlo sampling or numerical quadrature. An advantage of the Maximum Entropy formalism is that both can be calculated trivially after having completed the optimisation. To calculate the differential entropy we simply note that

$$\mathcal{S}(p) = \int p(\lambda)(1 + \sum_i^m \alpha_i x^i) d\lambda = 1 + \sum_i^m \alpha_i \mu_i \quad (14)$$

where $p(\lambda) = \exp[-(1 + \sum_i^m \alpha_i x^i)]$.

The KL divergence between two Maximum Entropy spectra, $p(\lambda) = \exp[-(1 + \sum_i \alpha_i x^i)]$ and $q(\lambda) = \exp[-(1 + \sum_i \beta_i x^i)]$, can be written as

$$\mathcal{D}_{kl}(p||q) = \int p(\lambda) \frac{p(\lambda)}{q(\lambda)} d\lambda = -\sum_i (\alpha_i - \beta_i)\mu_p^i, \quad (15)$$

where μ_p^i refers to the i-th moment constraint of the density $p(\lambda)$. Similarly the Jensen-Shannon divergence can be written as

$$\frac{\mathcal{D}_{kl}(p||q) + \mathcal{D}_{kl}(q||p)}{2} = \frac{\sum_i (\alpha_i - \beta_i)(\mu_q^i - \mu_p^i)}{2}, \quad (16)$$

where all the α_i and β_i are derived from the optimisation and the μ's are given from the stochastic trace estimation.

7 EXPERIMENTS

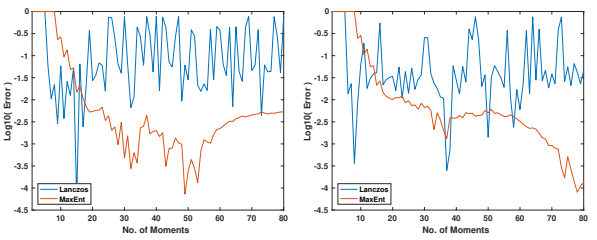
We use $d = 100$ Gaussian random vectors for our stochastic trace estimation, for both MaxEnt and Lanczos (Ubaru et al., 2017). When comparing MaxEnt with Lanczos we set the number of moments m equal to the number of Lanczos steps m_s i.e. $m = m_s$. We implement a quaderature MaxEnt algorithm 1. We use a grid size of 10^{-4} over the interval $[0, 1]$ and add diagonal noise on the Hessian to improve conditioning. 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 Laplacian with eigenvalues bounded by $[0, 2]$ and divide by 2. We use Python's in-built optimiser for both the MaxEnt lagrange multipliers and the parameters of the random networks of minimum divergence to our objective.

7.1 Synthetic Data

In order to test the robustness of the approach to networks with clusters of different structures, we implement a mixture of Erdos-Renyi, Watts-Strogatz and Barabsi-Albert networks using the Python package *NetworkX* and conduct multiple experiments using networks that have from 3 to 60 clusters, with each cluster containing 30 nodes. We connect the nodes between clusters randomly, with a single inter-cluster connection.

Algorithm 1 MaxEnt Algorithm

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



(a) 30 connected Clusters (b) 60 connected Clusters

Figure 3: Log Error of Community Detection using MaxEnt and Lanczos on Synthetic Data

We display the results for 30 clusters in Figure 3a and 60 clusters in Figure 3b. We see that 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 3a are unlikely to be replicated in real world experiments.

We observe a general improvement in performance for larger graphs, visible in the differences between figures 3a, 3b for MaxEnt and not Lanczos. This is to be expected as the true spectral density

$$p(\lambda) = \frac{1}{n} \sum_i^n \delta(\lambda - \lambda_i) \quad (17)$$

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

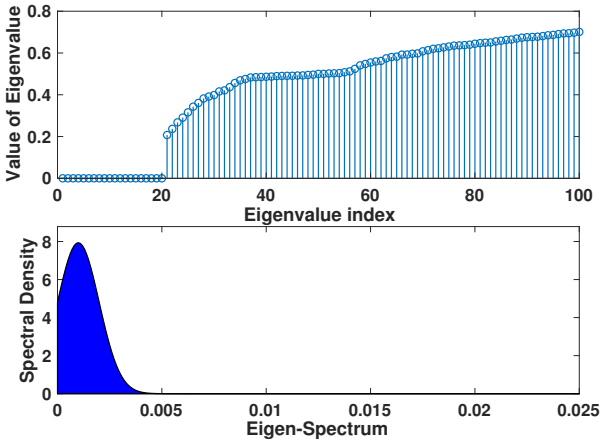


Figure 4: The smallest 100 Eigenvalues of the Email Dataset, with clear spectral gap along with the corresponding spectral density near the origin, showing a minimum at the value of the eigengap. The shaded area multiplied by the number of nodes n predicts the number of clusters.

space of solutions for the given constraints (Caticha, 2000).

7.2 Real Data

7.2.1 Small Real World

When the number of nodes $n \approx 10^4$, 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 the spectral gap near 0. We count 20 very small eigenvalues before a large jump in magnitude[5] as shown in Figure 4 and set this as the ground truth for the number of clusters in the network. 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, Astro-Physics, 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.

[5] measured in the log scale

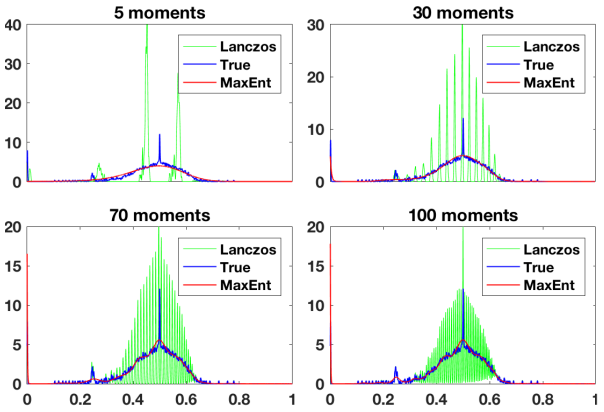
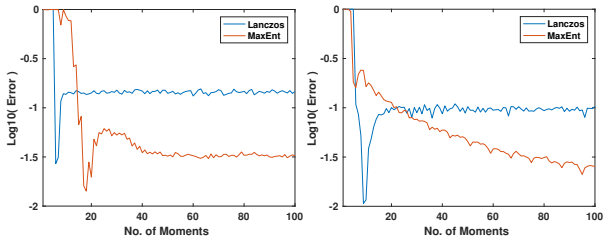


Figure 5: Spectral density for varying number of moments m, for both the MaxEnt and Lanczos algorithm as well as the ground truth.



(a) Email Dataset (b) NetScience Dataset

Figure 6: Log error of community detection using MaxEnt and Lanczos algorithms on for differing number of moments m.

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 5. 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 6a, 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 6 show that MaxEnt quickly outperforms the Lanczos algorithm after around 20 moments.

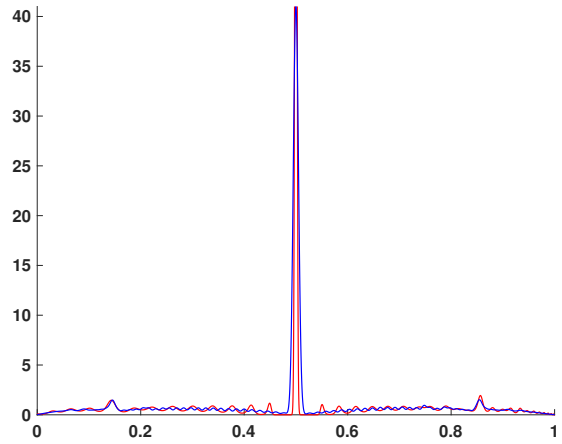


Figure 7: Spectral Density for Youtube Dataset $m = 150$, for MaxEnt and Lanczos Approximation

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. We note that the ground truth supplied in (Mislove et al., 2007a), regarding each connected component in a group as a separate ground-truth community and removing communities with less than 3 nodes is not universal. This definition, along with that of self-declared group membership (Yang and Leskovec, 2015), often leads to contradictions with our definition of a community, such as with the Orkut dataset, where the number of 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.

We show that our method continues to faithfully approximate the spectra of large graphs, as shown in Figure 7 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 3 for a varying number of moments.

Table 3: Cluster rrediction 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

Table 4: Average parameters estimated by MaxEnt for the 3 types of network

n	50	100	150
ERDOS-RENYI ($p = 0.6$)	0.600	0.598	0.6040
WATTS-STROGATZ ($p = 0.4$)	0.4683	0.4537	0.4129
BARABSI-ALBERT ($r = 0.4n$)	18.9360	40.2389	58.4275

7.2.3 Learning Real-world Network Types using MaxEnt and Jensen-Shannon Divergence

Determining which random graph models best fit real networks, characterised by their spectral divergence, so as to better understand their dynamics and characteristics has been explored in Biology (Takahashi et al., 2012). We replicate their synthetic experiments using our approximate MaxEnt spectrum, with the results reported in Table 4.

We generate random graphs of a given size n and parameters and then find its MaxEnt spectral characterisation. Then by generating other graphs of the same size, and running the Jensen-Shannon divergence between the original spectral density and the proposed through the inbuilt Python minimiser, we investigate whether one can recover the parameters of the input graph. We repeat the experiment for all 3 types of synthetic networks: Erdos-Renyi, Watts-Strogatz and Barabsi-Albert, and for each type, we repeat for different network sizes $n = 50, 100, 150$.

The results in Table 4 show that for randomly generated networks, given simply the approximate MaxEnt spectrum, we are able to rather well learn the parameters of the graph producing that spectrum.

We also conduct another set of experiments to test our method for robustness against changing spectral properties with network scale. We generate a Barabsi-Albert graph of $n = 5000$ with given parameters and then try to minimise the divergence between random graphs of different types, free parameters and fixed nodal size 1000. The results in Table 5 show that we successfully recover

Table 5: Minimum KL divergence between Entropic Spectrum of Youtube and that of Synthetic Networks

	SYNTHETIC	YOUTUBE
ERDOS-RENYI	2.662	7.728
WATTS-STROGATZ	7.6123	9.735
BARABSI-ALBERT	**2.001**	**7.593**

the correct type of the synthetic network.

As a real word example, we look for which random network among Erdos-Renyi, Watts-Strogatz and Barbasi-Albert can best model the YouTube dataset. We do this by minimizing the divergence between YouTube MaxEnt spectral density and those of the randomly generated graphs. We find that the Barbasi-Albert gives the lowest divergence, which aligns with other findings for social networks (Barabási and Albert, 1999).

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. We further show that spectral divergence techniques can be faithfully reproduced using our approximative methods.

The major advantage of the here presented algorithm using maximum entropy is its computational complexity which is $\mathcal{O}(n_{\text{nz}}md)$, where n_{nz} is the number of nonzeros 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 the eigengap heuristic and thus our definition of communities breaks down.

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.

A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

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.

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.

J. R. Hershey and P. A. Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–317. IEEE, 2007.

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.

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.

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.

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.

D. Y. Takahashi, J. R. Sato, C. E. Ferreira, and A. Fujita. Discriminating different classes of biological networks by analyzing the graphs spectra distribution. *PLoS One*, 7(12):e49949, 2012.

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.