Research Review

Clustering Signed Networks with the Geometric Mean of Laplacians [6]

R.E. Kusztos *rek43*

November 29, 2017

1 Introduction

Graph clustering is a key technique in unsupervised learning and has grown in popularity in recent years due to the collection of big network datasets, such as the social networks or brain networks. The techniques that have received a greater focus are mainly concerned with unsigned graphs. It has been suggested however that these ideas are not directly transferable to the problem of clustering in signed networks.

As such, [6] is presenting a novel model of spectral clustering applied to the problem of signed network. The paper is very proof-oriented, discussing how previous techniques are flawed when theoretically analysed on a stochastic block model (SBM) [8] for signed graphs, showing the advantage of the newly proposed method. Furthermore, *Mercado et al.* argue that this is the first application of the SBM in the spectral study of signed graphs.

Moreover, *Mercado et al.* propose a method to optimise the calculations used in the algorithm using Krylov subspaces. Their paper proposes the first optimisation fine-tuned for computing the geometric mean of sparse matrices [6]. This direction, however, will not be explored further in this review, the focus being on the applicability of the technique for graph clustering.

Lastly, through applications on real-world datasets, *Mercado et al.* show, for the first time [6], clustering structure in the Wikipedia RfA signed network [5].

2 Description

The vast majority of literature on spectral graph clustering is dealing primarily with positively weighted graphs. That is, for a graph given by a set of nodes N and edges E, and weight matrix A,

$$A = (a_{i,j})_{i,j \in N}, \quad a_{i,j} \geq 0.$$

The value of the weight is correlated with the similarity between the two nodes. It is natural to extend the framework to allow expression of 'dissimiliarities', i.e. both 'friend' or 'foe' relations. This is achieved via signed networks. Such datasets can be found in domains such as social media networks [4], but they can also be artificially created by combining a 'k-nearest neighbour' approach with a 'k-farthest-neighbour' as suggested by *Mercado et al.*.

2.1 Theoretical background

Spectral graph clustering is a method of finding communities in the nodes of a graph, by using methods of linear algebra. Given the weight matrix, A, we can compute the degree matrix, D, where

$$D_{i,i} = \sum_{j \in N \setminus \{i\}} w_{i,j}$$

Then, we can define the following matrices:

$$L = D - A \qquad \text{laplacian of a graph}$$
$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2} \qquad \text{normalised laplacian}$$

Using either of these laplacians, we can introduce the classical spectral graph clustering algorithm.

- Compute L (or L_{sym})
- $\Lambda \in R^n, V \in R^{n \times n} \leftarrow$ the eigenvalues and (normalized) eigenvectors of L, ordered by the magnitude of the eigenvalues.
- Select the smallest k $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_k \leq ...\text{in}\Lambda$ and the corresponding eigenvectors $U \in R^(n \times k)$
- k Means Clustering on U with features along the lines
- The output gives the labels of the n vertices

2.2 New matrices used for signed graphs

The paper proceeds by separating the positive and negative weights in two positive matrices as such:

$$W^+_{i,j} = w_{i,j} \quad \text{if} \quad w_{i,j} > 0 \quad \text{and 0 otherwise}$$
$$W^-_{i,j} = w_{i,j} \quad \text{if} \quad w_{i,j} < 0 \quad \text{and 0 otherwise}$$

It is argued that the W^+ is assortive, i.e. that the edges inside each cluster are higher in number than edges between different clusters, and hence [] the previously presented laplacian. Dually, the *signless laplacian* is introduced for the W^-.

$$Q = D + Q$$

Also, by using the notation:

$$D^+_{i,i} = \sum_{j=0}^{j=n} w^+_{i,j}$$
$$\bar{D}_{i,i} = \sum_{j=0}^{j=n} w^+_{i,j} + w^-_{i,j}$$

Some alternative laplacian matrices for using as input to the clustering algorithm 2.1. These will be used to evaluate the newly proposed method in Section 4.

$$L_{BR} = D^+ - W^+ + W^-$$
$$L_{BN} = \bar{D}^{-1} L_{BR}$$
$$L_{SR} = \bar{D} - W^+ + W^-$$
$$L_{SN} = \bar{D}^{-1/2} L_{SR} \bar{D}^{-1/2}$$
$$L_{AM} = L^+_{sym} + Q^-_{sym}.$$

2.3 Geometric means of Laplacians

Lastly, the geoemtric mean is introduced.

$$A\#B = A^{1/2}(A{-}1/2BA^{-1/2})^{1/2}A^{1/2}.$$

and, accordingly, the new measure used for clustering:

$$L_{GM} = L^{+}_{sym} \# Q^{-}_{sym}.$$

The paper carries on by introducing various theorems. These are relevant to both the proof that the method performs well on clustering tasks, by employing the probabilistic formalism and the stochastic block model, as well as additional theorems that are used in the optimisation part of the paper. In particular, it is proven than if the parameters of the SBM satisfy certain inequalities, the smallest eigenvalues of the geometric mean of the laplacian corresponds to a set of eigenvalues derived artificially to model the community structure of the SBM. (Theorem 3, [6]). This shows that the spectral clustering algorithm should theoretically yield a maximal result in this case.

Next, in Section 3, I will describe the implementation of the algorithm and the stochastic block model, followed by a reimplementation of the analysis performed by *Mercado et el.* combined with additional testing.

3 Methodology

3.1 Spectral clustering implementation

The implementation follows closely the one presented in algorithm 2.1. See Figure **??**.

By feeding different matrices into this algorithm we can reconstruct all the algorithms presented in the paper. Running all the algorithms under a profiler, we can see that the eigenvalue decomposition runtime trumps the rest of the algorithm, showing that optimisations at that step are very important.

3.2 Computing the matrices

The implementation of the matrices follows closely the mathematical formuli presented in section 3. The linear algebra operations are all performed using the numpy library.

The theoretical results require the argument of the classification procedure **??** to be positive definite matrices. Since both of the $LSim$ and $QSim$ can be proven to be positive semi-definite, we can make sure that the resulting value will be positive definite by adding a small value to the numbers. In the evaluation section, we will also explore how this value can alter the results computationally.

```python
import numpy as np
from sklearn.cluster import KMeans


def cluster_baseline(adj_matrix, k):
    n, m = adj_matrix.shape
    assert n == m

    eigvalues, eigvectors = np.linalg.eig(adj_matrix)

    ind = eigvalues.argsort()[:k]
    eigvalues = eigvalues[ind]
    eigvectors = eigvectors[:, ind]

    kmeans = KMeans(n_clusters=k)
    kmeans.fit(eigvectors)
    return kmeans.labels_
```

Figure 1: Spectral Clustering. In line 9, the numpy operation return normalized vectors. Since in most of my experiments, the calculations did not yield symmetric matrices, no optimisation could have been performed for computing the eigenvalues. A possible extension would be performing the highly parallelisabe operations on the GPU. Lines 11-13 select the k eigenvectors corresponding to the smallest eigenvalues. Lastly, the KMeans clustering algorithm is performed (15-17), by using the implementation provided in the scikit-learn package.

```python
def compute_LGM(W_plus, W_minus):
    n, m = W_plus.shape
    assert n == m

    LSim = get_normalized_laplacian(W_plus)
    LSim = up_shift(LSim, 0.001)

    QSim = get_normalized_signless_laplacian(W_minus)
    QSim = up_shift(QSim, 0.001)

    LGM = geometric_mean(LSim, QSim)
    return LGM
```

Figure 2: Computation of the LGM

3.3 Stochastic Block Matrices

Since the evaluation section will require a working implementation of the Stochastic Block Matrices(SBM), I will describe the implementation

presented in the original paper.
The SBM is a generative model for graphs, which focuses on the creation of graphs with communities in their structure. This model has been used before in performance discussions regarding spectral clustering [3], albeit on unsigned networks.

In order to extend the SBM model for signed networks, *Mercado et al.* pick a simplified model (this can also be found under the name *planted partition model*):
Firstly, we restrict the possible values of weights to 1, yielding a positive and a negative matrix. Secondly, we pick $p_i n^+ (p_i n^-)$ to be the probabilities of having a (negative) edge between nodes of the same cluster, and $p_o ut^+ (p_o ut^-)$ defined dually.

These constants are restricted by some inequalities that ensure various properties of the graph the SBM represents. However, those are presented mainly in the context of theorem proving, so we will instead focus on a working implementation in this section. In the evaluation section, we will analyse how breaking those inequalities affect the clustering.

We pick the arbitrary node to cluster allocation method of assigning every node n to cluster $n \ mod \ n_{vert}$. This ensures the clusters are equal, as required by the paper [6].
We instantiate two matrices, W^+ and W^- as the adjacency matrices of the positive and negative graphs respectively. Then for all i, j, if $i \equiv j \ (mod \ n_{vert})$ we set $W^*_{i,j} = W^*_{j,i} = 1$ with probability p^*_{in} and $W^*_{i,j} = W^*_{j,i} = 1$ with probability p^*_{out} where $* \in \{+, -\}$.Furthermore, we ensure that the matrix is symmetric by updating the values of $W_{i,j} = W_{j,i}$. Lastly, if $W^+_{i,j} = W^- i, j$, we set them to 0, as the graphs need to be defined correctly. See Figure **??** for a complete listing.

Picking correct values for an input matrix tends to be experimentally difficult. Small p-values relative to the number of nodes will yield disconnected components. This yields to null values in the diagonal matrix D, making it singular and breaking the computation tree.
On the other hand, big p-values yield graphs with an unclear clustering structure, for which custering scores are expected to be low.

4 Evaluation

The algorithms implemented in this work suffer from a complexitiy limitation which has been addressed in the actual papers. Therefore, I will focus

```python
import numpy as np
import numpy.random as rand


def stochastic_block_model(n_vert, n_clust, p_in_plus,
    p_in_minus, p_out_plus, p_out_minus):
    W_plus = np.zeros((n_vert, n_vert))
    W_minus = np.zeros((n_vert, n_vert))

    for i in range(n_vert):
        for j in range(i + 1, n_vert):
            if (i % n_clust == j % n_clust):
                if (rand.uniform() <= p_in_plus):
                    W_plus[i, j] = 1
                    W_plus[j, i] = 1
                if (rand.uniform() <= p_in_minus):
                    W_minus[i, j] = 1
                    W_minus[j, i] = 1
            else:
                if (rand.uniform() <= p_out_plus):
                    W_plus[i, j] = 1
                    W_plus[j, i] = 1
                if (rand.uniform() <= p_out_minus):
                    W_minus[i, j] = 1
                    W_minus[j, i] = 1
            if W_minus[i, j] == W_plus[i, j]:
                W_minus[i, j] = 0
                W_plus[i, j] = 0
                W_minus[j, i] = 0
                W_plus[j, i] = 0
    correct_labels = [x % n_clust for x in range(n_vert)]
    return W_plus, W_minus, correct_labels
```

Figure 3: SBM Graph Generation. The correct labels are returned as a convenience for the rest of the implementation.

on experiments with a lower number of nodes and edges.
I will first discuss the clustering method used, and then compare the algorithm with different variations on artificially generated graphs (SBM), as well as graph generated from standard datasets by using a k-nearest neighbours approach, as suggested by *Mercado et al.*

4.1 Clustering metric

To the best of my knowledge, the papers [6] doesn't specify exactly what is the clustering metric they have used, referring to it as simply *clustering error*. In the rest of this work, I use the *adjusted Rand score*. According to the sklearn documentation [7], *'The Rand Index computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings. The raw RI score is then adjusted for chance.'*

This score was chosen because it was a natural way to assess to success on clustering graphs generated by the SBM, where the community structure is clearly determined. Furthermore, it is easily interpreted. A value close to 0 suggest that the cluster assignment is close to random, whereas as a high (subunitary) score suggest correct assignment.

4.2 Evaluation of the SBM

The proofs presented as part of the paper make the assumption that the parameters of the SBM model satisfy a number of inequalities.

(E_+)	$p_{out}^+ < p_{in}^+$	(E_{vol})	$p_{in}^- + (k-1)p_{out}^- < p_{in}^+ + (k-1)p_{out}^+$
(E_-)	$p_{in}^- < p_{out}^-$	(E_{conf})	$(\frac{kp_{out}^+}{p_{in}^+ + (k-1)p_{out}^+})(\frac{kp_{out}^-}{p_{in}^- + (k-1)p_{out}^-}) < 1$
(E_{bal})	$p_{in}^- + p_{out}^+ < p_{in}^+ + p_{out}^-$	(E_G)	$(\frac{kp_{out}^+}{p_{in}^+ + (k-1)p_{out}^+})(1 + \frac{p^-in - p_{out}^-}{p_{in}^- + (k-1)p_{out}^-}) < 1$

Table 1: Conditions used in SBM analysis [6]

It is argued that the LBN, LSN and LAM should overperform the LGM in the cases when E_{bal} and E_{vol} hold simultaneously, whereas LGM should achieve a higher score in the cases where E_G holds. The paper suggests that the latter cases are more common, fact which lead to the discovery of cluster structure in the Wikipedia signed network by using LGM.

We analyse performance of the algorithms by picking parameters for the SBM that satisfy these axioms. In order to achieve consistent results, we will use a constant value of k. This decision can be argued by showing that there is an ideal ratio between the number of vertices and the number of clusters for which a cluster structure can be clearly identified in the model.

In the following, we pick a value of n = 1000 and k = 5.

Next, we randomly pick parameters that satisfy all the inequalities in table 1.

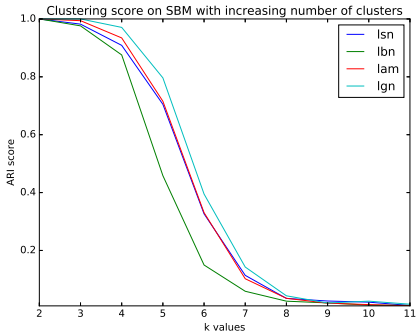


Figure 4: . The clustering score decreases as the number of clusters increases

p_{in}^{+}	p_{out}^{+}	p_{in}^{-}	p_{out}^{-}	LBN	LSN	LAM	**LGM**
0.025	0.02	0.03	0.075	0.792	0.451	0.783	**0.842**
0.005	0.01	0.015	0.05	**0.905**	0.869	0.883	0.902
0.01	0.05	0.04	0.07	0.0920	0.0694	**0.196**	0.160
0.005	0.045	0.045	0.06	0.235	0.124	0.407	**0.426**
0.005	0.005	0.01	0.055	0.985	0.980	0.936	**0.987**

Table 2: ARI score for clustering random graphs instantiated with the SBM model

This table presents the various algorithms and their performance with respect to the SBM model. The results are not at all conclusive, since the space has not been thoroughly explored (trying to do so would have a very high compute time).

Furthermore, although they had these properties, some instantiations of the SBM have yielded small results for all the algorithms, suggesting some limitations of this model to always create a clustering structure. Theoretical limits of this model are analysed in [1]

4.3 Runtime evaluation

The method is very computationally intensive. A timing graph of increasing node sizes is present below (table 5).

Running profiling techniques show that the slowest component in the code is the call to *np.linalg.eig*. The great merit of the paper is finding a way around having to compute eigenvalues of this matrix, relying instead on properties of the geometric mean. Unfortunately, exploring this is beyond

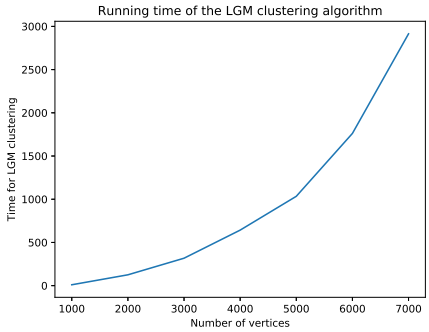


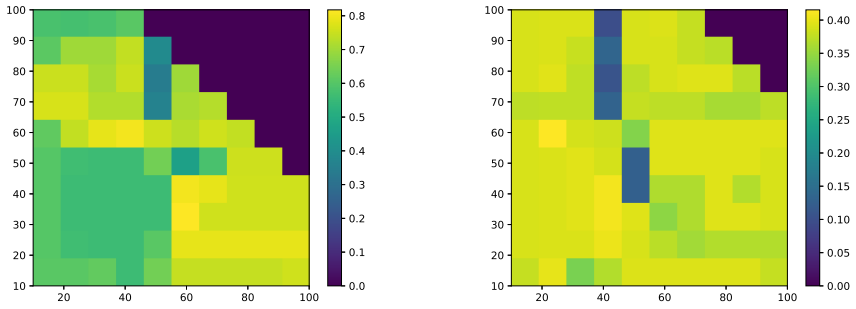
Figure 5: . Runtime of LGM clustering

the scope of this work.

4.4 Evaluation of standard datasets

In this subsection we analyse the performance of the algorithms on a standard dataset, present in the scikit-learn library. Since neither of these datasets are in a graph form, we need to apply a heuristic to transform the language of the problem. *Mercado et al.* suggest that we can generate the W_+ matrix by finding the k_+ nearest neighbours (using a metric distance such as the euclidean distance), and the W_- matrix by finding the k_- farthest neighbours (using e.g. the inverse of the euclidean)
A main observation is that the choice of k_+ and k_- is critical to the performance of any algorithm. To analyse this, we can compare the heatmaps of geometric mean of laplacian results for various values of k_+ and k_-. For the iris dataset, where obtain satisfying results (around 80%), there exists a correlation between the parameters. The same doesn't hold for the wine dataset.
Secondly, a possible question we can ask is whether having dissimilarity metrics (i.e. the edges relating farthest neighbours) actually improves the accuracy. To this end, I also include the calculation of standard spectral clustering using the laplacian, by only including k neighbours. Bellow, I present a table of the best results obtained by each of these algorithms for this dataset. The values are optimised by performing a grid search on possible values for k_+ and k_-.

We can compare these tables to graphs in the original paper. However, a conclusive comparison cannot be conducted because *Mercado et al.* do not specify, to the best of my knowledge, specifics of their clustering error



(a) Iris (b) Wine

Figure 6: Values of k^+ and k^- vs. LGM accuracy.

dataset(classes)	LSym	LSN	LBN	LAM	LGM
wine(3)	0.786	**0.818**	**0.818**	0.814	**0.818**
iris(3)	0.433	**0.448**	**0.448**	0.444	0.415
digits(10)	**0.805**	0.728	0.729	0.705	0.762
breast cancer(2)	0.635	0.670	0.670	**0.676**	0.534
ecoli(4)	0.751	0.679	0.679	**0.778**	0.694

Table 3: ARI for Datasets in scikit-learn toy examples (wine, iris, digits, and breast cancer) as well as URI(ecoli)

calculations. On these datasets, the papers report the ratio of times when a certain method was better or strictly better than the other, treating each selection of k as a separate problem. However, we treated the ks as hyperparameters to the machine learning model, and reported the best result via grid search. If instead we performed the original papers' analysis, we obtain slightly different results, we a favourable bias for the method that discards the farthest neighbour. This suggests that maybe, the authors used a differnt metric for finding the k nearest (farthest) neighbours.

4.5 SNAP datasets

The paper suggests that the LGM method has, for the first time, found community structure in the wikipedia data. Unfortunately, I have not been able to replicate this result. However, I have been able to output results in the bitcoin-otc and bitcoin-alpha data. [2]. One problem with analysing

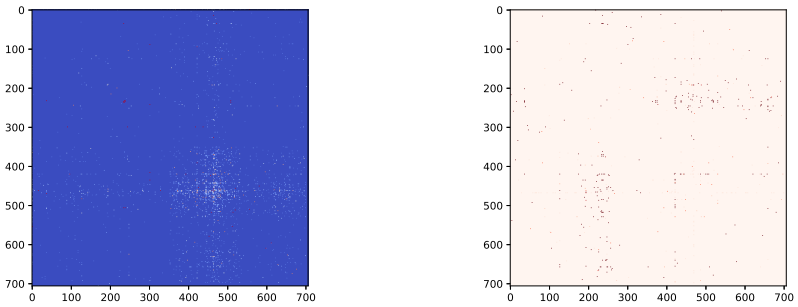
dataset / method	LSym	LSN	LBN	LAM	LGM
wine	**0.304**	0.240	0.144	0.141	0.171
iris	0.247	0.179	0.130	0.140	**0.305**
digits	**0.357**	0.107	0.143	0.143	0.250
breast cancer	**0.379**	0.207	0.103	0.276	0.034
ecoli	**0.407**	0.139	0.074	0.231	0.148

Table 4: Ratio of highest score during the grid search operation

graph data in this format is the presence of disconnected components. If there are disconnected components in either of the two matrices, W^+ or W^-, then their degree matrices will be singular. This breaks the computation tree by disabling computing the inverse.

An initial solution I applied to alleviate this problem is to recursively remove all the null rows and columns in the matrices. This makes sense in the current context since we are only interested in finding communities in connected components. I have not analysed the optimality of this method (whether the final matrix will have a maximal size), but it yielded good results.

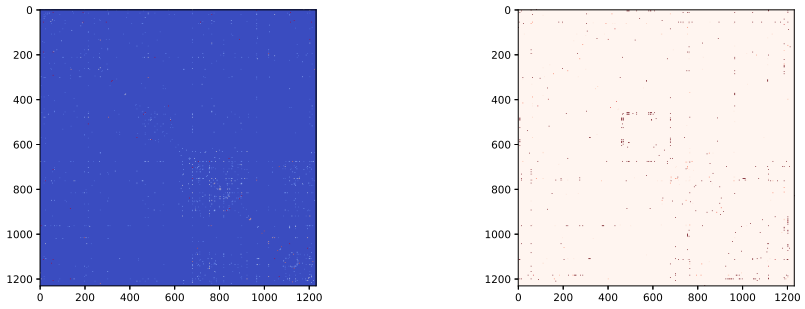
Bellow, I present the W^+ and W^- matrices for the two datasets, sorted after the LGM clustering.



(a) W^+ (b) W^-

Figure 7: Clustering on the Bitcoin-Alpha-Dataset [2].

Although the data is very sparse, communities can be clearly identified in the W^+. Unfortunately, I have not been able to reproduce heatmaps similar to those present in the paper.



(a) W^+ (b) W^-

Figure 8: Clustering on the Bitcoin-OTC-Dataset [2].

5 Conclusion

Mercado et al. [6] propose a new method for performing spectral clustering on signed graph. They address the problem of comparing to previous methods in a theoretical manner, and argue that the spectral clustering using LGM should perform better on more graphs than the others. Furthermore, they manage to overcome the challenge posed by an expensive eigenvalue decomposition by leveraging optimisations from properties of the geometric mean.

In the current work, I have tried to replicate their results and expand on some evaluation techniques. Firstly, I have tried to analyse the performance of clustering algorithms on the SBM in Section 4.2. In most of my tests, the LGM method yielded better results on SBM parametrized according to the axioms from table 1. Very bipolar outcome result from all the clustering methods, showing that the SBM method is not guaranteed to create communities.

Next, I analysed the standard datasets in section 4.4. Additionally to the methods used previously, I have also tried to compare signed graph clustering to unsigned graph clustering. Unfortunately, the latter resulted in better accuracy than any of the signed methods. The LGM method seems to perform neither better nor worse compared to the other signed methods. However, my results are not conclusive, since I was very limited by the high computational cost (see figure 5). I have also shown results of the grid search for the hyperparameters k^+ and k^-, which signal yet again that graph clustering tends to be, in general, limited by the vast differences

between graph structures.

Finally, I have tried to apply the method on a signed graph dataset, to this end choosing the Bitcoin Otc and Alpha datasets [2]. Although less clear than similar figures presented by *Mercado et al.*, communities have certainly been detected.

References

[1] Emmanuel Abbe and Colin Sandon. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 670–688. IEEE, 2015.

[2] Srijan Kumar, Francesca Spezzano, VS Subrahmanian, and Christos Faloutsos. Edge weight prediction in weighted signed networks. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 221–230. IEEE, 2016.

[3] Jing Lei, Alessandro Rinaldo, et al. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237, 2015.

[4] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1361–1370. ACM, 2010.

[5] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[6] Pedro Mercado, Francesco Tudisco, and Matthias Hein. Clustering signed networks with the geometric mean of laplacians. In *Advances in Neural Information Processing Systems*, pages 4421–4429, 2016.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[8] Karl Rohe, Sourav Chatterjee, Bin Yu, et al. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.