BLIND EXTRACTION OF EQUITABLE PARTITIONS FROM GRAPH SIGNALS

Michael Scholkemper, Michael T. Schaub

Department of Computer Science, RWTH Aachen University, Germany

ABSTRACT

Finding equitable partitions is closely related to the extraction of graph symmetries and of interest in a variety of applications context such as node role detection, cluster synchronization, consensus dynamics, and network control problems. In this work we study a blind identification problem in which we aim to recover an equitable partition of a network without the knowledge of the network's edges but based solely on the observations of the outputs of an unknown graph filter. Specifically, we consider two settings. First, we consider a scenario in which we can control the input to the graph filter and present a method to extract the partition inspired by the well known Weisfeiler-Lehman (color refinement) algorithm. Second, we generalize this idea to a setting where only observe the outputs to random, low-rank excitations of the graph filter, and present a simple spectral algorithm to extract the relevant equitable partitions. Finally, we establish theoretical bounds on the error that this spectral detection scheme incurs and perform numerical experiments that illustrate our theoretical results and compare both algorithms.

***Index Terms*—** Equitable partitions, Weisfeiler Lehman algorithm, spectral analysis, topology inference, graph symmetry

1. INTRODUCTION

Networks have become a powerful abstraction for complex systems [1, 2]. To comprehend such networks, we often seek patterns in their connections, which would allow us to comprehend the system in simpler terms. A common theme is to divide the nodes—and by extension the units of the underlying system—into groups of similar nodes. For instance, in the context of community detection, we consider nodes as similar if they are tightly-knit together, or share similar neighborhoods [3]. This notion of node similarity is thus bound to the specific position of the nodes in the graph, i.e., the identity of their neighboring nodes. In contrast, we may want to split nodes into groups according to whether they play a similar *role* in the graph [4], irrespective of their exact position. As an example, consider a division into hubs and peripheral nodes according to their degree, a split for which the exact identity of the neighboring nodes is not essential. While in this specific example defining a degree-similarity measure between nodes is simple, how to define a similarity measure between nodes in a position independent manner is a non-trivial question in general.

Rather than trying to identify similar nodes, many traditional approaches in social network analysis consider the definition of nodes roles based on *exact node equivalences*, such as regular equivalence or automorphic equivalence [5]. A specific form of such a partition into exact node equivalence classes is an *equitable partition* (EP), which may be intuitively defined recursively as sets of nodes that are connected to the same number of equivalent nodes. These EPs generalize orbit partitions related to automorphic equivalence and are thus closely related to graph symmetries. Knowledge of EPs can thus, e.g., facilitate the computation of network statistics such as centrality measures [6]. As they are associated to certain spectral signatures, EPs are also relevant for the study of dynamical processes on networks such as cluster synchronization [7, 8], consensus dynamics [9], and network control problems [10].

Motivated by this interplay between network dynamics and EPs, in this work we ask the question: *Can we detect the presence of an EP in a network solely based on a small number of nodal observations of a dynamical process acting on the network?* For this, we adopt a graph signal processing perspective [11], in which we model the dynamics as a graph signal filtered by an unknown filter representing the dynamics. Our task is then to recover the EP solely based on a small number of outputs of this filter.

Related work. Network topology inference has been studied extensively in the literature [12, 13]. Inferring the complete topology of a network can however require a large number of samples and may thus be infeasible in practice. A relatively recent development is the idea to bypass this inference of the exact network topology and directly estimate network characteristics in the form of community structure [14, 15, 16] or centrality measures [17, 18, 19] from graph signals. Learning graph characteristics directly in this way benefits from a better sample complexity, since only a low-dimensional set of spectral features must be inferred rather than the whole graph. This manuscript falls squarely within this line of work, but focuses on EPs as a different network feature, which results in some distinct challenges. Specifically, we cannot rely on the estimation of a dominant invariant subspace, but must estimate and select a subset of relevant eigenvectors from the whole spectrum of the graph.

Contributions and outline. We present two algorithms to tackle the problem of extracting the coarsest EP from the observation of graph signals under two different scenarios. First, we consider a scenario where we can control the input to the (unknown) graph filter, while having no access to the graph. We present an algorithm that exactly recovers the coarsest EP in this setting. Second, we consider a fully "blind" estimation problem, where we only have access to noisy, random, low-rank excitations of the graph filter. For this we derive a simple spectral algorithm and derive theoretical error bounds for its performance under certain assumptions. Finally, we illustrate our results and compare the two algorithms.

2. NOTATION

Graphs. A simple graph $G = (V, E)$ consists of a set of nodes V and a set of edges $E = \{uv \mid u, v \in V\}$. The neighborhood $N(v) = \{x \mid vx \in E\}$ of a node v is the set of all nodes connected to v. A graph is *undirected* if $uv \in E \iff vu \in E$. We may consider more general (non-simple) graphs that allow for directed edges, self loops $vv \in E$, or assign positive weights $w : E \to \mathbb{R}_+$

We acknowledge partial funding from the Ministry of Culture and Science of North Rhine-Westphalia (NRW Rückkehrprogramm) and the Excellence Strategy of the Federal Government and the Länder.

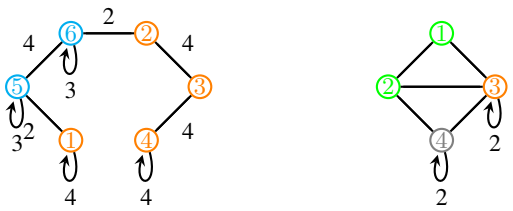

Fig. 1: Examples of graphs where the structural eigenvectors of the cEP behave irregularly. The depicted graphs are loopy to make the examples small. However, we can think of this as the condensed graphs A^π of some larger simple graph. (a) $[1, 1, 1, 1, 2, 2]$ is the perron vector of the graph. However, the cEP has 6 classes. (b) This graph also only has one EP, but the adjacency matrix is singular. Therefore, an eigenvector with eigenvalue 0 is part of the cEP.

to the edges of the graph, rendering it a *weighted graph*.

Matrices. For a matrix M, $M_{i,j}$ is the component in the i-th row and j-th column. We use $M_{i,\cdot}$ to denote the i-th row vector of M and $M_{\cdot,j}$ to denote the j-th column vector. The span of a matrix $M \in \mathbb{R}^{n\times m}$ is defined as the set $\text{span}(M) = \{x \in \mathbb{R}^n \mid x = Mv \text{ for some } v \in \mathbb{R}^m\}$. For a matrix X, we denote by $X \in \text{span}(M)$ if $X_{\cdot,j} \in \text{span}(M)$ for all j. $\mathbb{I}_n$ is the identity matrix and $\mathbb{1}_n$ the all-ones vector of size n respectively. Given a graph $G = (V, E)$, we identify the set of nodes with $\{1, \ldots, n\}$. An *adjacency matrix* of a given graph is a matrix A with entries $A_{u,v} = 0$ if $uv \notin E$ and $A_{u,v} = w(uv)$ otherwise, where we set $w(uv) = 1$ for unweighted graphs for all $uv \in E$.

Equitable partitions. A *partition* $\mathcal{C} = \{C_1, ..., C_k\}$ of a graph into k classes splits the nodes into disjoint sets C_i such that $C_i \cap C_j = \emptyset$ for $i \neq j$ and $V = \bigcup_i C_i$. An *equitable partition* (EP) is a partition such that within each class C_i the neighborhood of each node is partitioned into parts of the same size for all classes C_j. Formally, in an EP it holds for all nodes $v, u \in C_i$ in the same class that:

$$|\{N(v) \cap C_j\}| = |\{N(u) \cap C_j\}| \text{ for all } j \in \{1, ..., k\}. \quad (1)$$

A standard algorithm to detect EPs is the Weisfeiler-Lehman algorithm [20], a combinatorial algorithm which iteratively refines a coloring of the nodes until a stable coloring is reached. It finds the *coarsest EP* (cEP), meaning that with the fewest classes. Note that it is typically this cEP one aims to find, as it provides the largest reduction in complexity for describing the structure of the graph. Furthermore, any graph will have a trivial finest EP with $|V|$ classes.

3. EQUITABLE PARTITIONS AND EIGENVECTORS

In this section we recap how the presence of an equitable partition manifests itself in the spectral properties of the adjacency matrix. We will subsequently use these spectral properties to detect the presence of a cEP from the output of a graph filter.

Throughout the paper, we consider an undirected graph G with a cEP that has k classes encoded by the indicator matrix H with entries $H_{ij} = 1$ if $i \in C_j$ and $H_{ij} = 0$ otherwise. It now holds that

$$AH = H(H^\top H)^{-1}H^\top AH =: HA^\pi \quad (2)$$

where A^π is the adjacency matrix of the so-called *quotient graph* associated to the EP. Note that A^π is not necessarily simple nor undirected. The converse of eq. (2) also holds. If there exists an indicator matrix $H \in \{0, 1\}^{n\times k}$ as defined above and $AH = HA^\pi$, then the graph has an EP with k classes as indicated by H.

Spectral signatures of EPs The algebraic characterization of EPs in terms of Equation (2) has some noteworthy consequences. First, the eigenvalues of A^π are a subset of the eigenvalues of A. Second, the eigenvectors of A^π can be scaled up by H to become eigenvectors of A. Both of these statements are implied by the following argument: Let $A^\pi v = \lambda v$ define an eigenpair of A^π, then $AHv = HA^\pi v = \lambda Hv$ is an eigenpair of A.

Thus, A has k (unnormalized) eigenvectors of the form $z = Hv$ that are by construction block constant on the classes of the EP. We call these eigenvectors of A associated with the cEP *structural eigenvectors*. As A is symmetric, these eigenvectors form an orthogonal basis of a closed subspace. Moreover, as the structural eigenvectors are block-wise constant, the same subspace is also spanned by the indicator vectors of the classes, i.e., the columns of H.

This motivates the following intuitive idea to find the cEP *spectrally* by analyzing the eigenvectors of the adjacency matrix A: *Find the smallest number k of eigenvectors that are block constant on the same k blocks*. The vectors indicating the blocks of these eigenvectors will then correspond to the columns of the indicator matrix H for the cEP of the graph.

Since any structural eigenvector is of the form $z = Hv$, we may even hope to find the cEP by simply computing a single structural eigenvector, provided that the entries of the eigenvector v of the quotient graph are all distinct. Given such an eigenvector z, we could then simply read off the node-equivalences classes by checking whether or not two nodes are assigned the same value in the vector z. In fact, we can always identify at least one structural eigenvector of the cEP easily using the following proposition, which follows from the non-negativity of the adjacency and the algebraic characterization of the cEP given above.

Proposition 1. *If the Perron vector (dominant eigenvector) of a graph is unique, then it is a structural eigenvector.*

However, there are several obstacles to implement the above algorithmic idea. First, even if we are given a structural eigenvector of the cEP, simply grouping the nodes according to their entries in the eigenvector may not reveal the cEP. For instance, focussing only on the Perron vector may not yield the correct cEP, as nodes in different classes may still be assigned the same value. Second, the eigenvectors associated to the cEP may be associated to any eigenvalue of A, i.e., they are not necessarily the dominant eigenvectors of A.

Both of these problematic cases are shown in Figure 1. Figure Figure 1a provides an example where the dominant eigenvector (Perron vector) has fewer than k distinct entries. Figure Figure 1b shows an eigenvector of the cEP with small eigenvalues (in this case zero). While there area thus alwasys exactly k structural eigenvectors of the cEP, we must first determine which eigenvectors are indeed structural eigenvectors of the cEP and which are not.

4. SCENARIO I: BLIND BUT IN CONTROL

In the following, we consider a setup in which we cannot see the graph directly, but can sample the input/output behavior of a graph filter in form of a matrix polynomial $f(A) = \sum_k h_k A^k$, where $h_k \in \mathbb{R}$ are filter coefficients. To illustrate our main ideas, we will restrict ourselves for this section to the simplest case, where $f(A) = A$ and we can control the input to the filter. In Section 5 we will then concern ourselves with a fully "blind" cEP inference problem, where we have no control over the inputs.

Algorithm 1: BlindWL

$O \leftarrow$ oracle returning Ax;
$B \leftarrow \mathbb{1}_{|V|}$; ◁ *start with global partition indicator matrix*;
while $O(B) \notin \mathrm{span}(B)$ **do**
  for $i, j \in V$ **do**
    $B_{i,j} \leftarrow \begin{cases} 1 & \text{if } O(B)_{i,\cdot} = O(B)_{j,\cdot} \wedge B_{i,\cdot} = B_{j,\cdot} \\ 0 & \text{else} \end{cases}$
  end
  remove redundant columns from B
end
return B ◁ *return coarsest equitable partition*

For now let us assume we observe the outputs $y = Ax \in \mathbb{R}^n$ to a set of inputs x we can choose. Clearly, we could reconstruct the whole graph using sufficiently many inputs localized at single nodes, since $Y = A\mathbb{I}_n$. However, if we simply aim to identify a cEP with a relatively small number of classes, considerably fewer inputs suffice.

Our idea here is to use input/output behavior as an oracle to simulate the well-known Weisfeiler Lehman algorithm (WL) [20], also known as color refinement. Starting from an initial coloring $c^{(0)}$ at time $t = 0$ (usually the same color for all nodes), the WL algorithm updates the color of each node v iteratively as follows:

$$c^{(t+1)}(v) = \mathrm{hash}\left(c^{(t)}(v), \{\{c^{(t)}(x) \mid x \in N(v)\}\}\right)$$

where the doubled brackets denote a "multi-set", i.e. a set in which an element can appear more than once. Here the hash function is an injective function and ensures that nodes that have (i) the same color in previous iteration t and (ii) the same set of colors in their neighborhood, are assigned the same color in the next iteration $t+1$. Evidently, every step of the algorithm refines the coloring until at some point the partition of the graph induced by the colors stays the same. At this point, all nodes within the same class have the same number of neighbors to each class, i.e., the algorithm found an EP — the cEP to be precise [21]. Using the oracle for $y = Ax$, we present a "blind" version of the WL algorithm in Algorithm 1. A similar algorithm was proposed by [22] for the computation of fractional isomorphisms based on conditional gradients.

Properties of the blindWL algorithm. It is relatively easy to see that Algorithm 1 indeed finds the coarsest equitable partition. After termination, $AB \in \mathrm{span}(B)$, meaning that there exists some matrix A^π with $AB = BA^\pi$. Furthermore, consider the cEP represented by H^*. Throughout the execution, if $B = H^*B^\pi \in \mathrm{span}(H^*)$, then $AB = AH^*B^\pi = H^*A^\pi B^\pi \in \mathrm{span}(H^*)$. Since $\mathbb{1}_{|V|} = H^*\mathbb{1}_{|H^*|}$, B stays in $\mathrm{span}(H^*)$ and B eventually represents the same EP as H^*.

In fact, Algorithm 1 induces the same partitions as the WL algorithm in each iteration of the while loop: we start with the same color for all nodes, as encoded in the all ones vector. If the number of neighbors of a certain color c is different for two nodes u, v the WL algorithm will put them in two different classes. The corresponding components $(AB)_{u,c} \neq (AB)_{v,c}$ will also be distinct. Hence, in the next iteration, $B_{u,v} = 0$ meaning u and v are also put into distinct classes by the blind WL algorithm.

A benefit of the blindWL algorithm is that the intermediate row representations $(AB)_{i,\cdot}$ of a node i yield an embedding each iteration rather than colors that, if distinct, provide no method of comparison. For example one can cluster these embeddings to obtain an even coarser partition into nodes that are similar. This circumvents the sensitivity of the WL algorithm to minor perturbations in the graph: indeed, adding a single edge can disrupt an exact symmetry and yields a much finer cEP. While crucial to the original application of the WL (graph isomorphism checking), a more robust approach to assigning the classes is useful for node role extraction and the completely blind problem setting (section 6).

5. SCENARIO II: TRULY BLIND IDENTIFICATION OF EQUITABLE PARTITIONS

We now consider a scenario in which we aim to infer the cEP, but merely observe the outputs of a graph filter excited by a noisy, *random* low-rank excitation over which we have no control.

$$y = \alpha f(A)\tilde{H}x + (1-\alpha)z. \tag{3}$$

Here $x \sim \mathcal{N}(0, \mathbb{I}_k)$ and $z \sim \mathcal{N}(0, \mathbb{I}_n)$ are jointly Gaussian independent random vectors that are each sampled i.i.d from a normal distribution, $\alpha \in [0,1]$ is a parameter that regulates how strongly the structural eigenvectors are excited, and $\tilde{H} = H\,\mathrm{diag}(1/\sqrt{|C_i|})$. For simplicity, we assume that $f(A)$ and $f(A)^2$ have the same cEP indicated by H as A. Though this seems restrictive, for generic graphs most filters will fulfill this requirement. Indeed, the cEP of A is always an EP of A^k as well, though it may not be the coarsest.

Now observe that the covariance matrix of the above process has the following form:

$$\Sigma = \mathbb{E}[yy^T] = P\Gamma P^T = \alpha^2 f(A)\tilde{H}\tilde{H}^T f(A)^T + (1-\alpha)^2\mathbb{I}_n$$

Because $f(A)$ has an cEP as indicated by H, for any eigenvector $f(A^\pi)v = \lambda v$, associated to a structural eigenvector we have:

$$\Sigma H v = (\alpha^2\lambda^2 + (1-\alpha)^2)Hv$$

Hence, the structural eigenvectors of Σ are the same as the structural eigenvectors of $f(A)$, which are scaled-up, block constant eigenvectors of $f(A^\pi)$. Now, let $f(A) = V\Lambda V^T$ denote the spectral decomposition of the (symmetric) matrix $f(A)$, and denote by V_{EP} the subset of structural eigenvectors.

If we consider the cEP $\mathcal{C}^* = \{C_1^*, ..., C_k^*\}$ associated to $f(A)$ and define the k-means cost function:

$$F(\mathcal{C}, V) = \sum_{C\in\mathcal{C}}\sum_{i\in C}\left\| V_{i,\cdot} - \frac{1}{|C|}\sum_{j\in C} V_{j,\cdot}\right\|_2^2 \tag{4}$$

it is easy to see that $F(\mathcal{C}^*, V_{\text{EP}}) = 0$, as the eigenvectors of $f(A)$ are block-wise constant on the classes of the cEP.

Hence if we had access to (a good estimate of) the covariance matrix Σ, we could simply use k-means to find candidates for the cEP, provided we can supply the correct eigenvectors V to the algorithm. As the above calculations show, the parameter α regulates the scale of the eigenvalues associated to the structural eigenvectors. For sufficiently large values of α most structural eigenvectors will, in fact, be the dominant eigenvectors. Assuming that we know the number of classes k of the cEP to be found, we may thus simply pick the top k eigenvectors of Σ and optimize the k-means objective to obtain the blocks of the EP — a procedure akin to spectral clustering. Here we estimate the covariance matrix by the sample covariance based on sampled outputs y_i for $i \in 1, \ldots, s$:

$$\hat{\Sigma} = \frac{1}{s}\sum_{i=1}^{s} y_i y_i^T = \hat{P}\hat{\Gamma}\hat{P}^T$$

For this setup we can show the following result.

Theorem 1. *Let $\hat{\Sigma}-\Sigma=\Delta$ and let $y^1,...,y^s \in \mathbb{R}^n$ be independent samples from the graph filter as in eq.* (3) *and let $r = \mathrm{Tr}(\Sigma)/\|\Sigma\|_2$. Let the following conditions hold:*

1. *KMeans finds a partition $\hat{\mathcal{C}} = \{\hat{C}_1,...,\hat{C}_k\}$ that minimizes $F(\hat{\mathcal{C}}, \hat{P}_{EP})$.*
2. *$\|y^i\|_2^2 \leq K\mathbb{E}[\|y\|_2^2]$ is bounded almost surely.*
3. *There exists $\delta > 0$ s.t. $\|\Delta\| + \delta \leq \gamma_k - \hat{\gamma}_{k+1}$.*

Then, for $c > 0$ and with probability at least $1-c$:

$$\sqrt{F(\hat{\mathcal{C}}, V_{EP})} \leq \frac{\sqrt{8k}\,\|\Sigma\|_2\,\Theta\left(\sqrt{\frac{K^2 r\log(n/c)}{s}} + \frac{K^2 r\log(n/c)}{s}\right)}{\delta}$$

for some constant Θ.

The proof of the theorem can be found in the full version available here. It is inspired by [14, 23] and uses a concentration inequality and the Davis Kahan $\sin(\theta)$ theorem. The theorem itself bounds the error of the partition found by the simple spectral clustering algorithm. The consistency statement that in the limit $s \to \infty$ the error vanishes and the extraction of the cEP is exact immediately follows.

A similar statement can be made about an adjusted variant of the blindWL algorithm applied to the estimated covariance matrix $\hat{\Sigma}$. We can simply replace the exact equality conditions in the computation of the intermediate matrix B in Algorithm 1 with a clustering algorithm that allows for some variance. In the limit $s \to \infty$, the error of the approximate oracle $\hat{\Sigma}x$ goes toward 0: $\left\|\hat{\Sigma}x - \Sigma x\right\| = \|\Delta x\| \to 0$. Therefore, the adjusted blindWL algorithm also exactly recovers the cEP with infinitely many samples. In the next section we explore numerically how the two algorithms compare with finitely many samples.

6. EXPERIMENTS

In this section we perform some experiments that support the theoretical findings of this paper. Toward this end, we use the setup as described in section 5, eq. (3). While the proposed spectral algorithm of section 5 is fit for the task, the blindWL algorithm (1) must be altered slightly as indicated above. In particular, we no longer have control over the inputs, thus the oracle O for $y = Ax$ is replaced with an approximate oracle $\hat{y} = \hat{\Sigma}x$.

Accordingly, we use a (robust) version of the algorithm, in which we replace the exact equality check in line 5 in algorithm 1 and instead fit a gaussian mixture on the rows of $O(B)$. The adjusted algorithm then uses the indicator vectors of the found clusters as the new intermediate B. We note that in the scenario of section 4, both variants of the algorithm yield the same result (under the assumption that the clustering algorithm fits the data optimally).

The graphs used in our synthetic test are sampled from a locally colored configuration model [24]. As opposed to the original configuration model, in the locally colored configuration model, the edge stubs also specify what color the incident nodes should have. Specifically, each node in the model has two main sets of parameters: an assignment to a (color) class and a number of colored stubs with which the node is required to link to other classes (which amounts to specifying a partition indicator matrix H and a quotient graph A^π). Given that the desired constraints can be met, we obtain a simple graph without self-loops or multi-edges. Stated differently, using the locally colored configuration model, we can fix the number of colored neighbors for each node and thus guarantee that the sampled graph has an EP $AH = HA^\pi$. For more details we refer the reader to the code available here.

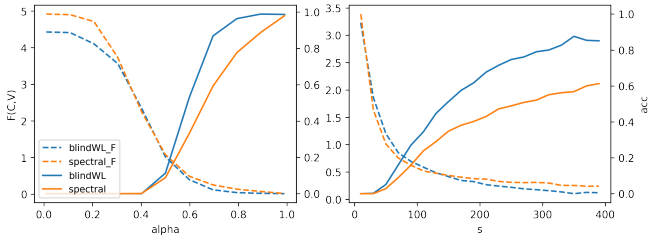

Fig. 2: **Graph-level and node-level performance of the algorithms.** The figure shows the graph-level accuracy on the right axis (continuous line) and the node-level cost function $F(\hat{\mathcal{C}}, V_{\text{EP}})$ on the left axis (dashed line) of each graph. It also shows the progression of these metrics with fixed sample size $s = 300$ and varying noise parameter α (left), and with fixed $\alpha = 0.7$ and varying s (right).

In the experiments, graphs with 300 nodes and an EP with 6 same sized classes were used. In each experiment, we randomly sampled a symmetric matrix A^π uniformly from the integers $\{0,...,4\}^{6\times 6}$. Subsequently, we sampled the matrix A, generated s outputs y^i (for $i \in 1,\ldots,s$) and evaluated the algorithms.

We measure the performance of both algorithms using graph-level accuracy, that is, an output partition receives a score of 1 if it is exactly equivalent to the planted partition; else the score is 0. Note that this is a quite strict measure, as a correct class assignment for all but one node is still counted as a complete failure to recover the EP. As a second, node-level measure, we use the cost function $F(\hat{\mathcal{C}}, V_{\text{EP}})$ as defined in eq. (4), which can give insight into the quality of wrong partitions. Both measures are reported as the mean score over 1000 repeated experiments for each of the parameter configurations shown in Figure 2.

In the right plot of Figure 2, a rapid decrease in the cost function and a slightly less steep increase in the accuracy can be seen for increasing sample size, which underlines our theoretical findings in Theorem 1. Though quite close in the node-level measure, the blindWL algorithm already has considerably higher accuracy using only few samples.

In the left plot of Figure 2 and with no noise at all, both algorithms find the correct partitions. However, the blindWL algorithm is again more robust when increasing the noise. The fact that the algorithms do not converge to the same score at $\alpha = 0$ can be explained by the distinct clustering methods. While KMeans always finds 6 clusters the gaussian mixture used in the blindWL can use less than 6 components in the mixture. This should also be kept in mind when comparing the two algorithms, as KMeans requires the number of classes as input, whereas the blindWL algorithm can infer the number of classes from the data.

7. CONCLUSION

We presented approaches to blindly extracting structural information in the form of an equitable partition of an unobserved graph from only such graph signals. In theorem 1, we established a theoretical bound on the error of such an inferred partition and went on to compare the spectral clustering and blindWL approaches experimentally. An interesting direction for future research may be to exploit this notion of node roles, e.g., the quotient structure may be employed for faster computations of certain graph filters.

8. REFERENCES

[1] Steven H Strogatz, "Exploring complex networks," *Nature*, vol. 410, no. 6825, pp. 268–276, 2001.

[2] Mark Newman, *Networks*, Oxford University Press, 2018.

[3] Santo Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[4] Ryan A Rossi, Di Jin, Sungchul Kim, Nesreen K Ahmed, Danai Koutra, and John Boaz Lee, "On proximity and structural role-based embeddings in networks: Misconceptions, techniques, and applications," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 14, no. 5, pp. 1–37, 2020.

[5] Ulrik Brandes, *Network analysis: methodological foundations*, vol. 3418, Springer Science & Business Media, 2005.

[6] Rubén J Sánchez-García, "Exploiting symmetry in network analysis," *Communications Physics*, vol. 3, no. 1, pp. 1–15, 2020.

[7] Louis M Pecora, Francesco Sorrentino, Aaron M Hagerstrom, Thomas E Murphy, and Rajarshi Roy, "Cluster synchronization and isolated desynchronization in complex networks with symmetries," *Nature communications*, vol. 5, no. 1, pp. 1–8, 2014.

[8] Michael T Schaub, Neave O'Clery, Yazan N Billeh, Jean-Charles Delvenne, Renaud Lambiotte, and Mauricio Barahona, "Graph partitions and cluster synchronization in networks of oscillators," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 26, no. 9, pp. 094821, 2016.

[9] Ye Yuan, G-B Stan, Ling Shi, Mauricio Barahona, and Jorge Goncalves, "Decentralised minimum-time consensus," *Automatica*, vol. 49, no. 5, pp. 1227–1235, 2013.

[10] Simone Martini, Magnus Egerstedt, and Antonio Bicchi, "Controllability analysis of multi-agent systems using relaxed equitable partitions," *International Journal of Systems, Control and Communications*, vol. 2, no. 1-3, pp. 100–121, 2010.

[11] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[12] Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst, "Learning laplacian matrix in smooth graph signal representations," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, 2016.

[13] Vassilis Kalofolias, "How to learn a graph from smooth signals," in *Artificial Intelligence and Statistics*. PMLR, 2016, pp. 920–929.

[14] Hoi-To Wai, Santiago Segarra, Asuman E Ozdaglar, Anna Scaglione, and Ali Jadbabaie, "Community detection from low-rank excitations of a graph filter," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4044–4048.

[15] T Mitchell Roddenberry, Michael T Schaub, Hoi-To Wai, and Santiago Segarra, "Exact blind community detection from signals on multiple graphs," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5016–5030, 2020.

[16] Michael T Schaub, Santiago Segarra, and John N Tsitsiklis, "Blind identification of stochastic block models from dynamical observations," *SIAM Journal on Mathematics of Data Science*, vol. 2, no. 2, pp. 335–367, 2020.

[17] T Mitchell Roddenberry and Santiago Segarra, "Blind inference of centrality rankings from graph signals," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5335–5339.

[18] Yiran He and Hoi-To Wai, "Estimating centrality blindly from low-pass filtered graph signals," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5330–5334.

[19] Yiran He and Hoi-To Wai, "Detecting central nodes from low-rank excited graph signals via structured factor analysis," *arXiv preprint arXiv:2109.13573*, 2021.

[20] Boris Weisfeiler and Andrei Leman, "The reduction of a graph to canonical form and the algebra which appears therein," *NTI, Series*, vol. 2, no. 9, pp. 12–16, 1968.

[21] Robert Paige and Robert E Tarjan, "Three partition refinement algorithms," *SIAM Journal on Computing*, vol. 16, no. 6, pp. 973–989, 1987.

[22] Kristian Kersting, Martin Mladenov, Roman Garnett, and Martin Grohe, "Power iterated color refinement," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[23] Hoi-To Wai, Santiago Segarra, Asuman E Ozdaglar, Anna Scaglione, and Ali Jadbabaie, "Blind community detection from low-rank excitations of a graph filter," *IEEE Transactions on signal processing*, vol. 68, pp. 436–451, 2019.

[24] Bo Söderberg, "Random graphs with hidden color," *Physical Review E*, vol. 68, no. 1, pp. 015102, 2003.

[25] Christos Boutsidis, Prabhanjan Kambadur, and Alex Gittens, "Spectral clustering via the power method-provably," in *International conference on machine learning*. PMLR, 2015, pp. 40–48.

[26] Charles F Van Loan and G Golub, "Matrix computations (johns hopkins studies in mathematical sciences)," 1996.

[27] Roman Vershynin, *High-dimensional probability: An introduction with applications in data science*, vol. 47, Cambridge university press, 2018.