ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

# Anomalous Cluster Detection in Large Networks with Diffusion-Percolation Testing

Corentin Larroche[1,2], Johan Mazel[1], and Stephan Clémençon[2]

1- French National Cybersecurity Agency (ANSSI)
51 boulevard de La Tour-Maubourg, 75700 Paris 07 SP - France

2- Télécom Paris, Institut Polytechnique de Paris - LTCI
19 place Marguerite Perey, 91120 Palaiseau - France

**Abstract**. We propose a computationally efficient procedure for elevated mean detection on a connected subgraph of a network with node-related scalar observations. Our approach relies on two intuitions: first, a significant concentration of high observations in a connected subgraph implies that the subgraph induced by the nodes associated with the highest observations has a large connected component. Secondly, a greater detection power can be obtained in certain cases by denoising the observations using the network structure. Numerical experiments show that our procedure's detection performance and computational efficiency are both competitive.

## 1 Introduction

Given an undirected graph with scalar observations attached to its nodes, an anomalous cluster can be defined as a connected subset of nodes carrying significantly high observations [1]. Numerous real-world applications (e.g. sensor networks, object detection in images or disease outbreak detection) have motivated extensive research on detecting such clusters. The standard approach relies on scan statistics [2]: given a score function quantifying how significant a potential cluster is, the maximum of this score function over the set of potential clusters is used as a test statistic to detect the existence of a significant cluster. Computing this maximum then becomes the main challenge, essentially reducing cluster detection to a combinatorial optimization problem. Although efficient algorithms can be designed to find exact or approximate solutions to this problem, explicitly looking for the most significant cluster remains a computational bottleneck, especially when dealing with large networks.

In contrast, we propose an optimization-free detection procedure relying upon the following intuition: when removing all nodes except those carrying the highest observations, the size of the largest remaining connected component should be small in the absence of a cluster. On the other hand, when a cluster is present, most of its nodes should remain in the thresholded graph, leading to a significantly larger connected component. While thresholding-based detection methods have already been studied [3, 4], our contribution is a more generic and powerful test. In particular, we use a message passing scheme to denoise the observations prior to thresholding, which can help make potential clusters stand out. The resulting procedure, called the Diffusion-Percolation test, is described in more detail and evaluated through numerical experiments in the next sections.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

## 2 Cluster Detection, Scan Statistics and Alternatives

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected and connected graph, where $\mathcal{V} = \{v_1, \dots, v_n\}$ denotes the set of nodes of $\mathcal{G}$ and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is its edge set. Let $\mathbf{A} \in \{0,1\}^{n \times n}$ denote the adjacency matrix of $\mathcal{G}$ and $\mathbf{M} = \mathbf{D}^{-1}\mathbf{A}$ denote its row-normalized counterpart (where $\mathbf{D}$ is the diagonal matrix whose $k$-th diagonal coefficient is the degree of $v_k$). We define $\Lambda$ as the set of subsets of $\mathcal{V}$ whose induced subgraph in $\mathcal{G}$ is connected. For each node $v_k \in \mathcal{V}$, let $X_k$ be a real-valued random variable attached to $v_k$. Consider the following hypothesis testing problem: let $\mathcal{N}(\mu, \sigma^2)$ denote the normal distribution with mean $\mu$ and variance $\sigma^2$, then the null hypothesis is defined as $H_0 : X_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and, for each $\mathcal{S} \in \Lambda$,

$$H_{\mathcal{S}} : \exists \mu > 0,\, \forall v_k \in \mathcal{V},\, X_k \overset{\text{ind.}}{\sim} \mathcal{N}\left(\mu \mathbb{1}_{\{v_k \in \mathcal{S}\}}, 1\right)$$

is one possible alternative (where $\mathbb{1}_{\{\cdot\}}$ is the indicator function of an event). The problem of cluster detection can then be formulated as

$$H_0 \quad \text{vs.} \quad H_1 = \bigcup_{\mathcal{S} \in \Lambda} H_{\mathcal{S}}.$$

Note that we are only interested in detecting the presence of a cluster, thus our procedure does not aim to reconstruct $\mathcal{S}$ after rejecting the null hypothesis.

The standard approach to this detection problem relies on scan statistics. It first introduces a scoring function $f : \Lambda \to \mathbb{R}$, which, in the Gaussian case considered here, is defined by

$$\forall \mathcal{S} \in \Lambda,\, f(\mathcal{S}) = \frac{1}{\sqrt{|\mathcal{S}|}} \sum_{v_k \in \mathcal{S}} X_k,$$

where $|\mathcal{S}|$ denotes the size of $\mathcal{S}$. Given a threshold $\theta \in \mathbb{R}$, the null hypothesis is then rejected if $\max_{\mathcal{S} \in \Lambda} f(\mathcal{S}) \geq \theta$. This condition indeed ensures the existence of a cluster in $\mathcal{G}$ which is significant at threshold $\theta$. The detection problem is then recast as a combinatorial optimization problem, namely maximizing $f$ over the class $\Lambda$. However, solving this optimization problem is computationally intensive, especially for large graphs. Therefore, most contributions on practical cluster detection focus on exact or approximate computation of the scan statistic through efficient algorithms. Several ideas have been explored, such as considering only a subset of the class $\Lambda$ [5] or solving a convex relaxation of the problem [6]. Generic optimization methods have also been applied (e.g. simulated annealing [7] or branch and bound algorithms [8]).

Despite these advances, explicitly looking for the most significant cluster remains intrinsically expensive. Some authors thus proposed alternative approaches: Sharpnack et al. [9] introduced a simple approximation of the scan statistic based on spectral properties of the Laplacian, and Langovoy and Wittich [3] designed a thresholding-based test relying on percolation theory, which was further studied by Arias-Castro and Grimmett [4]. Our work can be seen as an extension of these methods.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

## 3 The Diffusion-Percolation Test

We propose a two-step methodology to detect the existence of a cluster. First, the signal $X = [X_1, \ldots, X_n]^\top$ is denoised through a message passing scheme in order to make potential clusters more visible. A percolation-based test statistic is then computed and calibrated using bootstrap replications of the input signal.

*Denoising the Signal – Diffusion Step.* In order to make cluster detection easier, we first take advantage of the graph-structured nature of $X$ to reduce the noise. Specifically, the denoised signal $\tilde{X}$ is obtained by averaging the observation attached to each node with those of its neighbors, yielding

$$\tilde{X} = \frac{1}{2}(\mathbf{M} + \mathbf{I}_n)X, \tag{1}$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix. Intuitively, this transformation can be expected to smooth out discrepancies between adjacent observations, thus eliminating isolated anomalies while preserving clusters. Note that more sophisticated tools from the field of graph signal processing [10] could be used to this end, but they would incur higher computational costs which we seek to avoid here.

*Looking for a Cluster – Percolation Step.* Having computed the smoothed signal $\tilde{X}$, the next step is to look for traces of a potential cluster. To this end, define $v_{(1)}, \ldots, v_{(n)}$ as the nodes of $\mathcal{G}$ sorted in descending order of the smoothed observations (i.e. $\tilde{X}_{(1)} \geq \ldots \geq \tilde{X}_{(n)}$). Then, for $k \in \{1, \ldots, n\}$, let $\mathcal{G}_{(k)}$ be the subgraph of $\mathcal{G}$ induced by $\{v_{(1)}, \ldots, v_{(k)}\}$, and let $\mathcal{C}_{(k)}$ denote its largest connected component. Clearly, $\left|\mathcal{C}_{(1)}\right| = 1$ and, since $\mathcal{G}$ is connected, $\left|\mathcal{C}_{(n)}\right| = n$. What happens between these two extremes depends upon the presence of a cluster: under the alternative $H_{\mathcal{S}}$, the nodes inside of $\mathcal{S}$ should rank close to the top, hence they should form a large connected component in $\mathcal{G}_{(k)}$ for small values of $k$. In contrast, under $H_0$, high observations are not located in a specific region of $\mathcal{G}$, thus no large connected component should appear in the first thresholded graphs. Therefore, the size of $\mathcal{C}_{(k)}$ for small enough values of $k$ can be used as a test statistic. "Small enough" is here defined in a data-driven way: define

$$K = \min\left\{k \geq 2,\, \mathbb{E}_0\left[\left|\mathcal{C}_{(k)}\right|\right] \geq \sqrt{n}\right\}, \tag{2}$$

where $\mathbb{E}_0[\,\cdot\,]$ denotes the expected value under $H_0$. Then the test statistic is

$$T_{\mathcal{G}}(X) = \frac{1}{n(K-1)} \sum_{k=2}^{K} \frac{\left|\mathcal{C}_{(k)}\right| - \mathbb{E}_0\left[\left|\mathcal{C}_{(k)}\right|\right]}{\mathbb{V}_0\left[\left|\mathcal{C}_{(k)}\right|\right]^{1/2}}, \tag{3}$$

where $\mathbb{V}_0[\,\cdot\,]$ denotes the variance under $H_0$.

*Computation and Calibration of the Test Statistic.* Two problems have not been addressed yet: first, the statistic $T_{\mathcal{G}}(X)$ depends on a priori unknown expected values and variances. Secondly, deciding whether to reject $H_0$ based on

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

$T_{\mathcal{G}}(X)$ requires a calibration step: how likely is the test statistic to be at least as high under the null hypothesis? Both issues are addressed by generating $B$ bootstrap replications of $X$, denoted $\{X^1, \dots, X^B\}$. Each one of these is obtained by uniformly sampling $X_1^b, \dots, X_n^b$ with replacement from $X_1, \dots, X_n$. It is then denoised using Eq. 1, yielding a smoothed signal $\tilde{X}^b$ from which we derive a sequence $\mathcal{C}_{(1)}^b, \dots, \mathcal{C}_{(n)}^b$ of largest connected components. We then compute estimates for the unknown moments,

$$\forall k \in \{2, \dots, n\},\ \hat{\mu}_{(k)} = \frac{1}{B}\sum_{b=1}^{B}\left|\mathcal{C}_{(k)}^b\right| \text{ and } \hat{\sigma}_{(k)} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\left|\mathcal{C}_{(k)}^b\right| - \hat{\mu}_{(k)}\right)^2},$$

which are plugged into Eq. 2 and Eq. 3, leading to

$$\hat{K} = \min\left\{k \geq 2,\ \hat{\mu}_{(k)} \geq \sqrt{n}\right\}, \qquad \hat{T}_{\mathcal{G}}(X) = \frac{1}{n\left(\hat{K}-1\right)}\sum_{k=2}^{\hat{K}}\frac{\left|\mathcal{C}_{(k)}\right| - \hat{\mu}_{(k)}}{\hat{\sigma}_{(k)}}.$$

The estimated test statistic is finally used to compute the empirical $p$-value

$$\hat{p} = \frac{1}{B}\sum_{b=1}^{B}\mathbb{1}_{\left\{\hat{T}_{\mathcal{G}}(X^b) \geq \hat{T}_{\mathcal{G}}(X)\right\}}, \text{ where } \hat{T}_{\mathcal{G}}(X^b) = \frac{1}{n\left(\hat{K}-1\right)}\sum_{k=2}^{\hat{K}}\frac{\left|\mathcal{C}_{(k)}^b\right| - \hat{\mu}_{(k)}}{\hat{\sigma}_{(k)}}.$$

This bootstrap-based approach makes our test usable even when the null distribution of the observations is unknown, the only assumption being that they are independent and identically distributed under $H_0$.

In terms of computational cost, Eq. 1 can be computed in $\mathcal{O}(|\mathcal{E}|)$ operations, and the sequence $\mathcal{C}_{(1)}, \dots, \mathcal{C}_{(n)}$ is obtained using the Newman-Ziff algorithm [11] ($\mathcal{O}(n)$ complexity). The observations need to be sorted beforehand and all these operations are repeated $B+1$ times, leading to $\mathcal{O}(B(n \log n + |\mathcal{E}|))$ complexity.

## 4 Experiments

We now evaluate our procedure on a synthetic dataset, which is generated as follows: first, 50 random graphs of various sizes are sampled using the Kronecker graph model [12]. More specifically, a single generator matrix $\Theta$ =[0.9 0.5; 0.5 0.3] is combined with 5 different numbers of Kronecker product iterations ($i \in \{10, 11, 12, 13, 14\}$) to generate 10 graphs for each value of $i$. Only the largest connected component of each graph is kept, yielding a connected graph with approximately $2^i$ nodes. We then generate 50 normal signals and 50 anomalous signals for each graph, where each anomalous signal has elevated mean $\mu$ on a random connected subgraph containing a proportion $\delta$ of the nodes.

We compare our procedure (Diffusion-Percolation, abbreviated DP) with 3 baselines: the first one, called Percolation Only (PO), is the proposed test without the denoising step. The second one is the Upper Level Set scan statistic

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

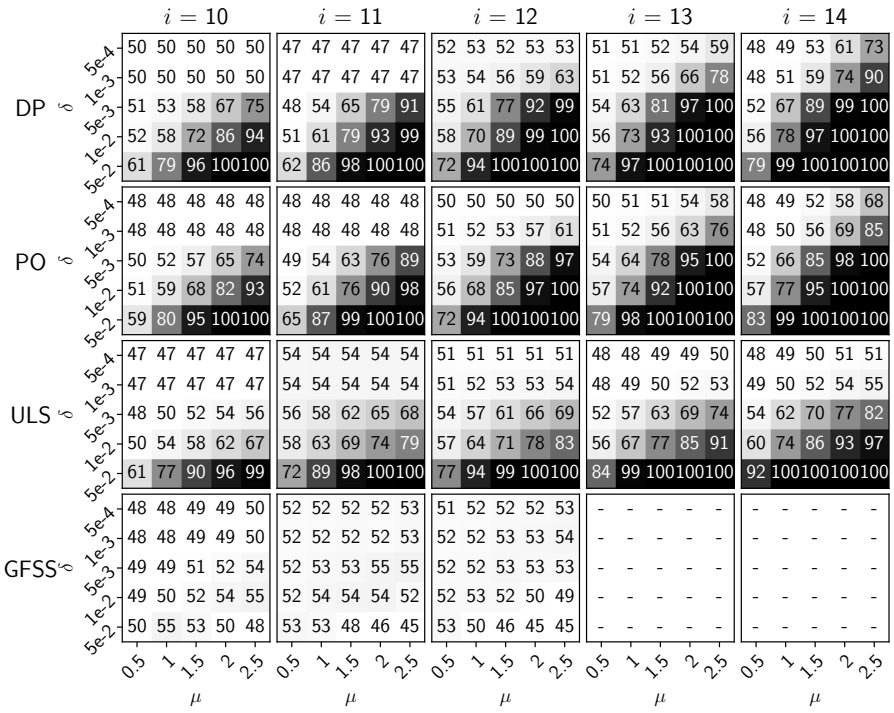


Fig. 1: Area under the ROC curve for each evaluated method, with different combinations of values of $i$, $\delta$ and $\mu$. Dashes indicate unavailable results due to excessive computation times.

(ULS [5]), which is an approximation of the scan statistic relying on a reduction of the search space. Finally, we include the adaptive Graph Fourier Scan Statistic (GFSS [9]), which approximates the scan statistic through the eigendecomposition of the Laplacian. DP, PO and ULS were implemented in Python, with the most intensive parts translated into C using Cython [13]. As for the GFSS, we used the open source Python implementation provided by the authors. Each test is calibrated using 1 000 simulations (bootstrap replications of the input for DP and permutations for other methods). Computations are run on a Debian 10 machine with 128GB RAM and a 2.2GHz, 20-core CPU.

The detection performance of each test is evaluated through the area under the Receiver Operating Characteristic (ROC) curve, and the results are displayed in Figure 1. DP and PO perform best overall, with DP doing slightly better for low $\delta$ and high $\mu$. This suggests that the diffusion step is especially useful when looking for small but strong clusters. This gain comes with a moderate loss in performance for low $\mu$ and high $\delta$, but ULS tends to perform best in this setting anyway. As for computation times, Figure 2 shows that DP, although slightly more expensive than ULS and PO, remains rather efficient, providing an interesting trade-off between detection performance and computational cost.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

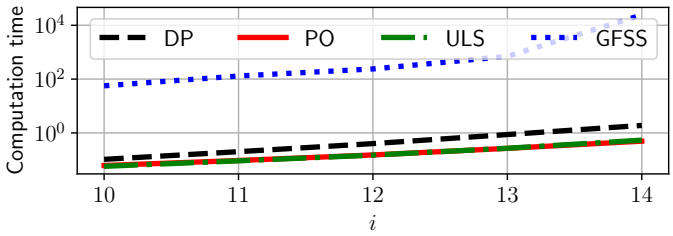


Fig. 2: Mean computation time (in seconds) for each evaluated method.

## 5 Conclusion

We propose an efficient and scalable statistical test for cluster detection in node-valued networks, and demonstrate its effectiveness through numerical experiments. Designing more sophisticated denoising schemes might be an interesting lead for future research. A higher detection power might also be obtained by looking for more complex anomalous patterns in the sequence $\left(\left|\mathcal{C}_{(1)}\right|, \ldots, \left|\mathcal{C}_{(n)}\right|\right)$, for instance by using functional anomaly detection methods.

## References

[1] Ery Arias-Castro, Emmanuel J Candes, and Arnaud Durand. Detection of an anomalous cluster in a network. *Ann. Stat.*, 39(1), 2011.

[2] J. Glaz, J. Naus, and S. Wallenstein. *Scan Statistics*. Springer, 2001.

[3] Mikhail Langovoy and Olaf Wittich. Robust nonparametric detection of objects in noisy images. *J. Nonparametr. Stat.*, 25(2), 2013.

[4] Ery Arias-Castro and Geoffrey R Grimmett. Cluster detection in networks using percolation. *Bernoulli*, 19(2), 2013.

[5] Ganapati P Patil and Charles Taillie. Upper level set scan statistic for detecting arbitrarily shaped hotspots. *Environ. Ecol. Stat.*, 11(2):183–197, 2004.

[6] Jing Qian and Venkatesh Saligrama. Efficient minimax signal detection on graphs. In *NeurIPS*, 2014.

[7] Luiz Duczmal and Renato Assuncao. A simulated annealing strategy for the detection of arbitrarily shaped spatial clusters. *Comput. Stat. Data Anal.*, 45(2), 2004.

[8] Skyler Speakman, Edward McFowland III, and Daniel B Neill. Scalable detection of anomalous patterns with connectivity constraints. *J. Comput. Graph. Stat.*, 24(4), 2015.

[9] James Sharpnack, Alessandro Rinaldo, and Aarti Singh. Detecting anomalous activity on networks with the graph fourier scan statistic. *IEEE Trans. Signal Process.*, 64(2), 2015.

[10] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proc. IEEE*, 106(5):808–828, 2018.

[11] Mark EJ Newman and Robert M Ziff. Fast monte carlo algorithm for site or bond percolation. *Phys. Rev. E*, 64(1), 2001.

[12] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker graphs: An approach to modeling networks. *J. Mach. Learn. Res.*, 11(2), 2010.

[13] Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn, and Kurt Smith. Cython: The best of both worlds. *Comput. Sci. Eng.*, 13(2):31–39, 2010.