Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference

A Graph-Theoretic Clustering Methodology Based on Vertex Attack Tolerance

Jeffrey Borwey[a] and **Darla Ahlert**[a] and **Tayo Obafemi-Ajayi**[b] and **Gunes Ercal**[a,*]
[a]Computer Science Dept., Southern Illinois Univ. Edwardsville, IL
[b]Electrical & Computer Eng. Dept., Missouri S & T, Rolla, MO
[*]Correspondence: gercal@siue.edu

Abstract

We consider a schema for graph-theoretic clustering of data using a node-based resilience measure called vertex attack tolerance (VAT). Resilience measures indicate worst case (critical) attack sets of edges or nodes in a network whose removal disconnects the graph into separate connected components: the resulting components form the basis for candidate clusters, and the critical sets of edges or nodes form the intercluster boundaries. Given a graph representation G of data, the vertex attack tolerance of G is $\tau(G) = \min_{S \subset V} \frac{|S|}{|V-S-C_{max}(V-S)|+1}$, where $C_{max}(V-S)$ is the largest component remaining in the graph upon the removal of critical node set S. We propose three principal variations of VAT-based clustering methodologies: hierarchical (hier-VAT-Clust), non-hierarchical (VAT-Clust) variations, and variation partial-VAT-Clust. The hierarchical implementation yielded the best results on both synthetic and real datasets. Partial-VAT-Clust is useful in data involving noise, as it attempts to remove the noise while clustering the actual data. We also explored possible graph representations options, such as geometric and k-nearest neighbors, and discuss it in context of clustering efficiency and accuracy.

Introduction and Related Work

Graph theoretic techniques for clustering are important not only when the data is given in network form, but also due to the established effectiveness of various graph partitioning techniques upon assigning a graph representation to the input data (Xu and Wunsch 2009; 2005). In graph theoretic contexts, the clustering problem is often represented within an optimization framework involving finding a k-partitioning of the vertices of the graph such that the *cuts* between groups are *sparse* and there exist additional constraints governing the relative sizes of each group (Shi and Malik 2000; Alpert, Kahng, and Yao 1999). Most commonly, the *bi-partitioning* problem is considered recursively as a basis for hierarchical clustering: Find the sparsest cut disconnecting the graph (into two groups), and continue finding the sparsest cut within a component until k components result. This variation of the sparsest cut problem is often taken interchangeably with problem of finding the *conductance* of a graph. Combinatorial conductance or edge based conductance is defined as

$$\begin{aligned}\Phi(G) &= \min_{S \subset V, Vol(S) \leq Vol(V)/2} \{ \tfrac{|Cut(S,V-S)|}{Vol|S|} \} \\ &= \min_{S \subset V, Vol(S) \leq Vol(V)/2} \{ \tfrac{|Cut(S,V-S)|}{\delta_S |S|} \}\end{aligned}$$

where $|Cut(S, V - S)|$ is the size of the cut separating S from $V - S$, $Vol(S)$ is the sum of the degrees of vertices in S, and δ_S is the *average* degree of vertices in S. Although conductance, multi-way cuts, and sparsest cuts are each hard to approximate to any constant factor (Chawla et al. 2006), relationships between eigenvectors and eigenvalues of the matrix representation of graphs and conductance of graphs has formed the basis of well known spectral approaches to graph partitioning (Shi and Malik 2000; Alpert, Kahng, and Yao 1999).

Another popular graph theoretic partitioning algorithm, presented in the context of community detection in social and biological networks, is the Girvan-Newman algorithm (Girvan and Newman 2002). It is based on a greedy removal of the highest betweenness edges until k connected components (clusters) results. The removed edges may be viewed as heuristic approximations of candidate sparse cuts between the resulting components, particularly in extremel scenarios involving edges with very high betweenness centrality. The Girvan-Newman algorithm appears to give meaningful results for certain social and biological networks, and has an advantage of simplicity of computation (if betweenness values are updated accurately and efficiently at each iteration). On the other hand, we are unaware of any previous work in which the Girvan-Newman algorithm has been applied to general datasets that were not originally already in network form.

It may be observed that common approaches to graph theoretic clustering, such as those mentioned above, solve a *resilience* problem on the graph while simultaneously outputting the connected components resulting from the removal of a *critical edge set* as the set of clusters. Specifically, the types of resilience problems considered thus far in the context of clustering are edge-based resilience problems, such as sparsest cut and conductance, which involve finding a critical edge set whose attack causes the greatest "disconnection" in the network. The duality between clustering and resilience is indicated by noting that critical sets of edges or nodes are those that lie on cluster boundaries, such that the components induced by their removal form a partial (in the case of node removals) or complete (in the case of edge removals) clustering of the nodes. Our central claim in this work is that *node based resilience measures* in general and *vertex attack tolerance* (VAT) (Ercal and Matta 2013; Matta, Borwey, and Ercal 2014) in particular may also be used effectively to cluster data and additionally address more generalized semi-clustering problems as well. The vertex attack tolerance of an undirected, connected graph $G = (V, E)$ is denoted $\tau(G)$ and defined as $\tau(G) = \min_{S \subset V}\{\frac{|S|}{|V-S-C_{max}(V-S)|+1}$, where $C_{max}(V-S)$ is the largest connected component in $V-S$. We focus on VAT in this work although some of our ideas and methods may also be applied to other node-based resilience measures such as vertex expansion(Louis, Raghavendra, and Vempala 2013; Feige, Hajiaghayi, and Lee 2005), integrity(Barefoot, Entringer, and Swart 1987), and toughness(Chvatal 2006). The VAT based clustering discussed in this paper is completely different from VAT - Visual Assessment Tendency clustering tool discussed in (Bezdek and Hathaway 2002). By VAT in this paper, we imply Vertex Attack Tolerance.

We have a number of motivations for considering vertex attack tolerance for generalized clustering problems. One primary motivation is that the following bounds relating VAT to both conductance and spectral gap have been proven for the case of regular degree graphs(Ercal 2014; Matta, Borwey, and Ercal 2014), indicating similar expected results between VAT-based clustering and spectral clustering for almost-regular graphs: For any d-regular connected graph $G = (V, E)$ with λ_2 the second largest eigenvalue of G's normalized adjacency matrix,

$$\frac{1}{d}\Phi(G) \leq \tau(G) \leq d^2\Phi(G) \quad (1)$$

Moreover,

$$\frac{\tau(G)^2}{2d^4} \leq 1 - \lambda_2 \leq 2d\tau(G) \quad (2)$$

However, homogeneous degree distributions are certainly not guaranteed, and in highly degree variant graphs there is a large discrepancy between vertex attack tolerance and conductance. Yet, VAT appears to partition the subgraph of G induced by the removal of the critical nodes S very well, with $G - S$ comprising several components for many network types such as scale-free models (Matta, Borwey, and Ercal 2014). Certainly, a fundamental difference between node and edge based resilience is that removal of a node of degree d can immediately result in up to d new components, whereas removal of an edge can disconnect at most one new component. Thus, we initially hypothesized that, whereas sparsest cuts and Girvan-Newman require input information on the desired number of clusters k, we might be able to directly use a single VAT computation to cluster the data accurately. Although our experimental results thus far have not confirmed this hypothesis, as we demonstrate, we shall later discuss a weighted generalization of VAT that may hold potential in this regard. Interestingly, the recursive, hierarchical implementation of VAT has been most fruitful in yielding high quality clustering results when applied to the experimental datasets.

Another fundamental difference between edge and node based resilience is the assignment of the set of *critical nodes*. When a complete clustering result is applicable, we assign these critical nodes to clusters using a greedy heuristic that prefers clusters with more neighbors. More fundamentally, however, we asked the question: Are there situations in which some nodes do not uniquely map to a set of clusters but rather exist naturally *between* clusters, either belonging to multiple clusters or to none at all? The semi-clustering approaches for Google Pregel(Malewicz et al. 2010) are motivated by considering that the former is a natural situation. As for the situation in which a node may not be in any cluster, such a node is simply *noise*. Thus, we have also tested the variation of our VAT-based clustering with no critical node reassignment for synthetic, noisy datasets. We discovered that this variation of VAT not only clusters such synthetic datasets well, but also cleans up some of the noise in the process, and rarely removes a non-noise node.

In the following sections, we detail our methods, results, conclusion, and future work.

Preliminaries for VAT-based Clustering

The following notations are useful:

$$\tau_S(G) = \frac{|S|}{|V-S-C_{max}(V-S)|+1}$$

so that clearly $\tau(G) = \min_{S \subset V} \tau_S(G)$ and correspondingly

$$S(\tau(G)) = argmin_{S \subset V} \tau_S(G)$$

Consider the connected components $\{C_i\}$ that result from the removal of the nodes $S(\tau(G))$, denoted by the *semi-partition* (Ligeza and Szpyrka 2007) of G *with respect to* τ as

$$SP(G) = \{C_i | C_i \text{ is a connected component of the subgraph of } G \text{ induced by } V - S(\tau(G))\}$$

Prior to proceeding to critical node assignment strategies that result in a complete partitioning of V, we wish to further note that even the most naive assignment strategy that groups each critical node with *every* component to which it is adjacent, already results in a *semi-clustering*(Malewicz et al. 2010). Whereas a semi-partitioning may violate the cover property of a partition while satisfying the disjointness property, a semi-clustering may violate the disjointness property of a partition while requiring that each element is covered by some group. For an incomplete clustering problem in which not all nodes should be assigned to a cluster, the straightforward use of VAT is to output each component of $SP(G)$ as a separate cluster. However, for the complete clustering problem, we must define a *critical node assignment strategy*, a mapping $f : S(\tau(G)) \rightarrow SP(G)$ of the critical nodes to components such that each extended component $C_i \cup \{f^{-1}(C_i)\}$ is a candidate cluster.

To extend a semi-partition to a complete clustering, we should restrict ourselves only to assignment strategies (and corresponding partitions) respecting adjacency relationships in the original graph. Prior to stating how we do this, for any $v \in V$ and $C \subset V$ let $\delta(v, C)$ denote the number of nodes in C to which v is adjacent. For convenience, also let $\delta(v) = \delta(v, V)$ simply denote the degree of node v. We present the following locally optimal critical node assignment schema:

Definition 0.1 (Maximal Neighboring Component) ***Critical Node Assignment Strategy:*** *The fundamental idea behind this Maximal Neighboring Component (MNC) critical node assignment schema is to assign each critical node $v \in S(\tau(G))$ to the component C in the semi-partition $SP_\tau(G)$ to which it has the most adjacencies, that is the one maximizing $\max_{C \in SP(G)} \delta(v, C)$, allowing ties. There are different ways to implement this strategy based on whether the critical nodes are considered to take turns in sequence. The* ***naive MNC*** *strategy simply assigns each critical node v to the component in the original semi-partition $SP_\tau(G)$ maximizing v's adjacencies, allowing ties. The* ***greedy sequential MNC*** *strategy, on the other hand, involves upkeeping a max-priority queue of critical nodes v keyed by the maximum value of their component neighbors $\delta(v, C)$ (over all components), then allowing the maximum such v to be assigned first, subsequently updating the neighbor relations based on the new extended components, then assigning the next maximum critical node, and so forth.*

Methods and Results

We implemented three variations of VAT-based clustering: partial-clustering with VAT, clustering with VAT, and hierarchical clustering with VAT. We refer to these as partial-VAT-Clust, VAT-Clust, and hier-VAT-Clust respectively. Like any graph-theoretic clustering algorithm, all variations require that the data first be transformed into a graph. We have implemented two graph types for the transformation: geometric graph and k-nearest neighbors (kNN) graph. Both are parametrized graph types, with radius threshold r required for the geometric graph and number of nearest neighbors chosen, k, required for the kNN graph. VAT requires that the input graph be connected, so the minimum values of r and k that may be considered are those achieving connectivity, below which the graph would be disconnected. We refer to this as the minimum connectivity regime (min-conn). While our implementation allows the choice of any above-connectivity r or k, the preliminary experiments suggest that the min-conn r and k often yield better results. The VAT implementation involves betweenness centrality computations which take $O(VE)$ time, thus for efficiency, the min-conn regime is also preferred to above connectivity regimes. In a similar vein, whereas one graph type is not preferrable to the other for all types of inputs (Maier, Luxburg, and Hein 2009), the min-conn kNN graphs tend to have significantly fewer edges than their min-conn geometric counterparts, thus making kNN more efficienct. Our kNN implementation does not enforces k-regularity but rather ensures that each node is connected to its k nearest neighbors, which need not be a symmetric relationship. Computation of the VAT measure itself is NP-hard to compute, hence we implement an approximation. We initialize a candidate attack set S_0 based on nodes with high be-

```
VAT Approximation
begin
  S := betweenessApprox(G);
  S := hillclimb(G, S);
where
proc betweenessApproxG) ≡
  n := size(G)/2;
  τ_best := 1.0;
  S_best := S_cur := ∅;
  do if n = 0 then exit fi;
     BC[] := BrandesBetweenessCentrality(G);
     V_max := max(BC);
     S_cur := S_cur ∪ V_max;
     if τ(S_cur) < τ_best
       then S_best := S_cur; τ_best := τ(S_cur);
     fi;
     G := G − V_max;
     n := n − 1;
  od.
proc hillclimb(G, V, S) ≡
  n := 0;
  do if n = size(G) then exit fi;
     if τ(G, flip(S, V_n)) < τ(G, S)
       then S := flip(S, V_n); n := 0;
        else n := n + 1;
     fi;
  od.
proc flip(S, V) ≡
  if V ∈ S then S := S − V;
             else S := S + V; fi.
end
```

Figure 1: Local Search Method to Approximate VAT

tweenness centrality, and use hill-climbing to attempt convergence to the actual critical set $S(\tau(G))$ that minimizes $\frac{|S|}{|V-S-C_{max}(V-S)|+1}$. Pseudo-code for our VAT computation is given in Figure . The asymptotic time complexity of the algorithm is $O(|V|(|V|^2 \log |V| + |V||E|) + H|E|)$, where H is the number of hill-climbing steps. Empirically, we have observed that $H = O(|V|)$.

Now we describe partial-VAT-Clust, VAT-Clust, and hier-VAT-Clust. Partial clustering with VAT involves transforming the input data points P into either a min-conn graph G, with both geometric and kNN options, computing the vertex attack tolerance $\tau(G)$, and then outputting each component $C_i \in SP(G)$ resulting from the removal of the critical nodes $S(\tau(G))$ as the clusters. VAT-Clust is similar to partial-VAT-Clust in its graph transfomation and VAT computation, except that VAT-Clust additionally involves a critical node assignment strategy for $S(\tau(G))$ to extend $SP(G)$ to a complete partition. We implemented the naive MNC strategy, as previously described, for VAT-Clust.

The hierarchical VAT-based clustering takes an input parameter of the minimum number of desired clusters K and fixes the graph type to be kNN. Hier-VAT-Clust initially runs VAT-Clust on the original graph G_0, checks if the resulting number of clusters (components extended by the critical

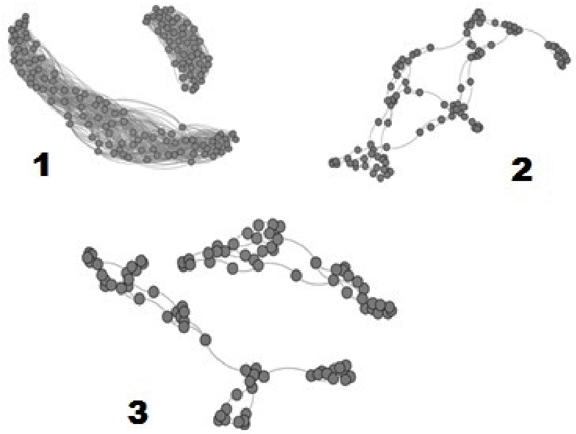

Figure 2: Hier-VAT-Clust visualization for Iris Dataset

nodes) is at least K. If so, we are done. If not, new min-conn kNN graphs $\{G_C\}$ are computed *for each of the current clusters* C. VAT values $\tau(G_C)$ are then computed for every new G_C. And, in the next recursive iteration of hier-VAT-Clust, VAT-Clust is used to further partition the cluster-graph G_{min} that has the lowest VAT value over all existing cluster-graphs, namely $G' = argmin\tau(G_C)$. This process of using VAT-Clust to subdivide the cluster-graph and achieving the lowest VAT value over all existing cluster-graphs is recursively repeated until the total number of clusters has reached at least K.

From our preliminary results on running the above algorithms, we observed that hier-VAT-Clust often needed to subdivide into more than 3 clusters. A single iteration of VAT-Clust usually did not suffice on the datasets. Therefore, all the following clustering results are based on running hier-VAT-Clust.

We used both synthetic and real datasets in our experiments. The real datasets were obtained from the UCI Machine Learning repository (Frank and Asuncion 2011) while the synthetic datasets were a subset of the synthetic datasets generated by Arbelaitz et al. (Arbelaitz et al. 2013) in their extensive clustering comparative experiments. These publicly available synthetic datasets were created to cover all the possible combinations of five factors: number of clusters (K), dimensionality, cluster overlap, cluster density and noise level. The subset we used contained no overlap i.e. strict. Table 1 summarizes our comparative results for the four real datasets: iris, breast-Wisconsin, ecoli, and wine. We compare hier-VAT-Clust to both k-means and the Girvan-Newman algorithm implemented on the min-conn kNN graph representation. Table 2 summarizes our results for synthetic datasets of K Gaussian mixtures with no overlap and no noise. Dim indicates the dimensionality, eq-dense indicates datasets with equal size clusters (100 nodes each), and uneq-dense indicates datasets in which one cluster is four times (400 nodes) as large as any of the other clusters (100 nodes each). Tables 3 and 4 summarize our bi-partitioning results for synthetic datasets with 10 percent noise and no overlap. All accuracy results are given as percent accuracy and computed based on all possible assignments of the computed clusters to the actual ground-truth labels.

Table 1: Accuracy Comparisons

Dataset	hier-VAT-Clust	k-means	G-N
iris	96	89.33	96.67
breast-WI	81.20	85.41	76.80
ecoli	76.49	58.43	60.71
wine	72.47	61.42	71.35

Table 2: Hier-VAT-Clust accuracy for synthetic datasets with no noise and no overlap

Dim	K	eq-dense	uneq-dense
2	2	99.5	100
2	4	99.25	57.57
2	8	99.5	100
4	2	99.5	100
4	4	99.5	99.86
4	8	99.86	99.82
8	2	99.5	100
8	4	99.5	99.86
8	8	99.63	100

Conclusion

We have presented a methodology for using a node based resilience measure, vertex attack tolerance, to perform a complete or partial clustering of data. Our results indicate that the hierarchical implementation hier-VAT-Clust performs very well for complete clustering of synthetic datasets without overlap and relatively well for the real datasets iris, ecoli, breast-WI, and wine as well. The 96 percent accuracy with which hier-VAT-Clust clusters the iris dataset indicates that VAT-based clustering is not limited by linear separability. In the cases of both iris and wine datasets, the differ-

Table 3: Partial-VAT-Clust accuracy for noisy synthetic datasets with $K = 2$, equal density, and no overlap

Dim	Graph type	Accuracy	non-noise removal
2	Geometric	100	0 out of 2
2	kNN	83	2 out of 5
4	Geometric	100	0 out of 4
4	kNN	99.5	1 out of 3
8	Geometric	100	0 out of 4
8	kNN	100	0 out of 2
2	Geometric	100	0 out of 7
2	kNN	100	0 out of 2
4	Geometric	100	0 out of 2
4	kNN	100	0 out of 1
8	Geometric	100	0 out of 2
8	kNN	100	0 out of 1

Table 4: Partial-VAT-Clust accuracy for noisy synthetic datasets with $K = 2$, unequal density, and no overlap

Dim	Graph type	Accuracy	non-noise removal
2	Geometric	100	0 out of 7
2	kNN	100	0 out of 2
4	Geometric	100	0 out of 6
4	kNN	99.4	3 out of 6
8	Geometric	100	0 out of 6
8	kNN	100	0 out of 5

ence between the accuracy of hier-VAT-Clust and Girvan-Newman algorithms is less than one and a half percent. Recall that Girvan-Newman may be viewed as an edge-based resilience clustering algorithm akin to techniques based on conductance or spectral gap, and constant approximation bounds exist between VAT and conductance for regular degree graphs. Therefore, it is reasonable to expect classes of examples in which Girvan-Newman and hier-VAT-Clust would behave similarly, though none of the generated graphs were exactly regular. In the case of the ecoli dataset, hier-VAT-Clust does significantly better than both k-means and Girvan-Newman. And, although k-means is the clear winner for the breast-WI dataset, hier-VAT-Clust also clearly beats Girvan-Newman in this case.

Regarding complete clustering results for hier-VAT-Clust on the synthetic dataset results, aside from the $Dim = 2$, $K = 4$, unequal density dataset which yielded only 57.57 accuracy, all other cases are almost perfectly clustered. The second synthetic datasets for $Dim = 2$ and $K = 4$ with unequal density yields over 99 percent accuracy, so the bad clustering achieved for the first dataset is not due to the setting of those parameters. The partial clustering results for partial-VAT-Clust are entirely perfect for bi-partitioning with the geometric graph representation and nearly perfect for bi-partitioning with the kNN graph representation in all cases except for the two dimensional equal density dataset which achieved an accuracy of 83 percent. Moreover, only in three cases were any non-noise removed because they comprised part of the critical node set.

Ongoing and Future Work

While we have demonstrated encouraging results for the four real datasets considered as well as synthetic datasets without overlap, we did not obtain such positive results while running hier-VAT-Clust on other real datasets and synthetic datasets involving overlap. Understanding the behavior of VAT-based clustering depending on properties of the dataset considered is the primary crux of ongoing and future work. As mentioned previously, there is a sensitivity to the graph representation as well that we continue to explore despite the increased computational overhead involved in denser graph representations. Notably, any computational overhead involved in VAT computations is further augmented in hierarchical VAT-based clustering due to the need to compute VAT *on every cluster-graph*. However, because VAT involves a hard optimization problem, the accuracy of VAT itself must be considered in any modification to the VAT computation. We continue to research possible improvements to our VAT algorithm and are concurrently testing a simulated annealing based approach. Additionally, of course, the critical node assignment strategy affects the overall performance of VAT-based clustering independently of the VAT computation. And, implementing the greedy sequential MNC critical node assignment strategy as well as hill-climbing optimizations based on internal validation indices are also being considered.

A particularly interesting question we wish to explore in future is: What is the minimum VAT corresponding to a given ground-truth clustering $\hat{P}$? Formally, given a particular partitioning $\hat{P}$ of the vertex set V into clusters $C \in \hat{P}$, what is the minimum set of critical nodes $\hat{S}$ whose removal may induce partitioning $\hat{P}$ (by some additional critical node assignment strategy)? Note that this requires the critical node set to lie adjacent to cluster boundaries, so that the minimization of $\hat{S}$ involves a vertex cover problem on the K-partite graph induced by the inter-cluster boundary edges. This also involves an NP-hard problem, but would yield insight into how well the VAT measure itself relates to an actual clustering quality. Moreover, since the actual VAT computation is approximate, we don't know a priori whether or not $\tau(\hat{S}) = \frac{|\hat{S}|}{|V - \hat{S} - C_{max}(V - \hat{S})| + 1}$ will be lower than our computed VAT value in some cases. This would suggest that the VAT algorithm can be improved for better clustering performance. It would be informative to know how relatively small or large $\tau(\hat{S})$ generally turns out to be, and for which classes of datasets. In general, this direction of research fundamentally yields information on how the VAT measure itself relates to clustering, independent of its implementation.

A more immediate line of research is variations to our hierarchical VAT-based clustering implementations. For example, although we plan to implement the partial version of hierarchical VAT clustering for noisy data, we may prefer to use a geometric graph representation due to the 100 percent accuracy for this representation for the bi-partitioning of noisy data. However, geometric graph representation tends to be denser, a problem that significantly augments the overhead of hierarchical implementations. Thus, we must consider graph representation and VAT implementation simultaneously with the extension of hierarchical VAT-Clust into partial clustering.

Regarding hierarchical VAT-Clust, we wish to understand why the hierarchical version of VAT-based clustering was needed for $K > 3$ instead of a single iteration of VAT-Clust, which we originally hypothesized may suffice. We plan to analyze the dendograms corresponding to the hierarchical implementation to better understand the recursion depth needed as a function of the datasets. However, what is clear is that VAT acts rather frugally in terms of size of the attack set S, preferring smaller S. This is probably due to the graph representation. There could exist graph representations of the same dataset that results in a large number of components even from the removal of very small S. Hence, the study of graph representations is again fundamental. Nonetheless, we will consider directly altering the frugality of the VAT measure itself by generalization of VAT to (α, β)-VAT, $\tau_{\alpha,\beta}(G) = \min_{S \subset V} \frac{\alpha|S|+\beta}{|V-S-C_{max}(V-S)|+1}$ as described in (Ercal 2014). Specifically, any of the $(1, \beta)$ parametrizations of VAT for $\beta \geq 1$ will allow larger critical sets S if doing so results in a more severe partitioning (or more desired cluster). Thus implementing and testing $(1, \beta)$-VAT-Clust is another direction of future work.

Finally, we have implemented all aspects of this work into a Graphical User Interface (GUI), including the graph generation as well as both VAT-based clustering and other clustering algorithms considered. We plan to continue the GUI development such that a user can run the graph generation and choice of clustering algorithm for various parametric settings. The GUI also outputs the graphs in GML format to facilitate viewing with graph visualization tools such as Gephi. Changes in VAT-based clustering implementations involve simultaneous changes to the GUI, and, additional functionality is continuously incorporated to our GUI along with plans to move it to an open-source platform.

Acknowledgement

We would like to thank Professor Donald Wunsch II for his helpful comments and guidance in writing this paper. We would also like to thank John Matta for helpful discussions in the early stages of this work.

References

Alpert, C. J.; Kahng, A. B.; and Yao, S.-Z. 1999. Spectral partitioning with multiple eigenvectors. *Discrete Applied Mathematics* 3–26.

Arbelaitz, O.; Gurrutxaga, I.; Muguerza, J.; PéRez, J. M.; and Perona, I. n. 2013. An extensive comparative study of cluster validity indices. *Pattern Recogn.* 46(1):243–256.

Barefoot, C.; Entringer, R.; and Swart, H. 1987. Vulnerability in graphs-a comparative survey. *Journal of Combinatorial Mathematics and Combinatorial Computing* 1:12–22.

Bezdek, J. C., and Hathaway, R. J. 2002. Vat: A tool for visual assessment of (cluster) tendency. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, volume 3, 2225–2230. IEEE.

Chawla, S.; Krauthgamer, R.; Kumar, R.; Rabani, Y.; and Sivakumar, D. 2006. On the hardness of approximating multicut and sparsest-cut. *Computational Complexity* 15(2):94–114.

Chvatal, V. 2006. Tough graphs and hamiltonian circuits. *Discrete Mathematics* 306(1011):910 – 917.

Ercal, G., and Matta, J. 2013. Resilience notions for scale-free networks. In *Complex Adaptive Systems*, 510–515.

Ercal, G. 2014. On vertex attack tolerance of regular graphs. *CoRR* abs/1409.2172.

Feige, U.; Hajiaghayi, M.; and Lee, J. 2005. Improved Approximation Algorithms for Minimum-Weight Vertex Separators. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, 563–572. ACM, New York.

Frank, A., and Asuncion, A. 2011. Uci machine learning repository, 2010. URL http://archive. ics. uci. edu/ml.

Girvan, M., and Newman, M. E. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99(12):7821–7826.

Ligeza, A., and Szpyrka, M. 2007. A note on granular sets and their relation to rough sets. In Kryszkiewicz, M.; Peters, J. F.; Rybinski, H.; and Skowron, A., eds., *RSEISP*, volume 4585 of *Lecture Notes in Computer Science*, 251–260. Springer.

Louis, A.; Raghavendra, P.; and Vempala, S. 2013. The complexity of approximating vertex expansion. *CoRR* abs/1304.3139.

Maier, M.; Luxburg, U. V.; and Hein, M. 2009. Influence of graph construction on graph-based clustering measures. In *Advances in neural information processing systems 21*, 1025–1032. Red Hook, NY, USA: Max-Planck-Gesellschaft.

Malewicz, G.; Austern, M. H.; Bik, A. J.; Dehnert, J. C.; Horn, I.; Leiser, N.; and Czajkowski, G. 2010. Pregel: A system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, 135–146. New York, NY, USA: ACM.

Matta, J.; Borwey, J.; and Ercal, G. 2014. Comparative resilience notions and vertex attack tolerance of scale-free networks. *CoRR* abs/1404.0103.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22(8):888–905.

Xu, R., and Wunsch, II, D. 2005. Survey of clustering algorithms. *Trans. Neur. Netw.* 16(3):645–678.

Xu, R., and Wunsch, D. 2009. *Clustering*. Wiley-IEEE Press.