Path-sharing: A new betweenness measure for community identification in networks

Paul McCarthy
Department of Computer Science, University of Otago, Dunedin New Zealand

***Abstract*—Community identification in networks has a wide range of practical applications, including data clustering and social network analysis. We present *path-sharing*, a new measure of betweenness, for use in identifying densely connected clusters in networks. We show that path-sharing performs well at identifying communities in artificial benchmark networks, giving performance comparable to that of state-of-the-art community identification techniques. We also demonstrate a practical use of path-sharing when used in community identification, by applying it to an image segmentation problem.**

I. Introduction

Discovering structural properties of networks has been a primary focus in complex network analysis for a number of years. The problem of identifying densely connected regions within large networks has its roots in the NP-complete problem of graph-partitioning (e.g. [1]) and social network analysis [2], yet the heterogeneous nature of networks means that an all-encompassing solution has not yet been found.

Newman and Girvan [3], [4] laid the groundwork for a large body of work, by formalising an approach to community identification in networks, drawing on statistical clustering techniques, and the concept of betweenness in graphs. This paper proposes a new measure of betweenness, termed *path-sharing*, which performs well at identifying community structure in a variety of networks. We have used Girvan and Newman's [3] and Lancichinetti et. al.'s [5] artificially clustered networks as a basis for comparison with the standard measure of betweenness in community identification, Newman and Girvan's edge-betweenness. We also compare path-sharing with Duch and Arenas' extremal optimization [6], and with Rosvall and Bergstrom's [7] random walks technique (referred to as *Infomap*), both proven high performance community identification algorithms.

Community identification is used in a wide range of areas, and has long been applied to the problem of image segmentation (e.g. [8], [9], [10]). For example, Shi and Malik's Normalized Cuts algorithm [11] segments an image by modelling it as a weighted graph, and searching for a optimal partitioning of vertices. We thus end the paper by demonstrating that path-sharing can be applied to such problems with promising results.

Throughout the remainder of this paper, the terms *graph* and *network* are used interchangeably. The term *cluster* is used to denote a subgraph, within a larger graph, consisting of vertices which are more densely connected to each other than to surrounding vertices [2]. The term *community* is used interchangeably with *cluster*. We have dealt exclusively with undirected graphs although, like edge-betweenness (e.g. [12]), path-sharing could easily be extended to directed variants.

II. Community Identification

Community identification in graphs is analogous to hierarchical clustering techniques, in statistical cluster analysis [3]. In the same way that hierarchical clustering divides, or combines, data points into groups based upon some measure of similarity, a community identification algorithm will divide (combine) a graph into (from) components, based upon some measure of betweenness between connected vertices. This paper is only concerned with *divisive* clustering [13], whereupon edges are successively removed from the graph; this causes the graph to break into smaller and smaller components, which represent the densely connected communities that were found. This is in contrast to *agglomerative* clustering, where the algorithm begins with a fully disconnected graph; edges are successively added between vertices, causing components to form, and eventually connect to each other.

A key issue, when using either method, is the question of when to stop the iterative process of adding/removing edges. Newman [4], [14] demonstrates that the *modularity* measure gives a good indication of the point at which an optimal community structure has been reached. The first step in the calculation of modularity is the creation of a symmetric $k \times k$ matrix $\mathbf{e}$, where k is the number of communities that have been found. Each element e_{ij} in the matrix $\mathbf{e}$ is simply the number of edges (in the original graph) which lie between communities i and j. The modularity may then be calculated by:

$$Q = \sum_i e_{ii} - a_i^2 \tag{1}$$

where $a_i = \sum_j e_{ij}$, i.e. the sum of all values in row i.

The divisive clustering algorithm used in this paper, as introduced by Girvan and Newman [3], [4], begins with the graph under analysis, with all edges intact, and proceeds as follows:

1) Calculate a betweenness measure for every edge in the graph (either edge-betweenness or path-sharing).
2) Remove the edge with the most extreme betweenness value (the highest value for edge-betweenness, or the lowest value for path-sharing). If multiple edges have the same value, randomly select one for removal.
3) Calculate the modularity on the original graph, with communities defined by the components created in the

current graph. Stop when the maximum possible modularity value has been reached.

Extremal optimization, introduced by Duch and Arenas [6], is a divisive techinque which, instead of using a measure of betweenness, uses heuristics to find a division of the graph that gives a near optimal modularity value. Extremal optimization is generally considered to give accurate results, at reasonable computational cost, and is known to perform very well on Girvan and Newman's benchmark networks [15]. Rosvall and Bergstrom's Infomap [7] uses the problem of finding an efficient coding scheme for a random walk on a network, as a proxy for determining the important structures, or communities, in the network. Infomap has been shown to perform very well on Lancichinetti et. al.'s benchmark graphs [16]. These two techinques, combined with the two classes of artificial networks, thus form a good basis for comparison of techniques for community identification.

To compare these techinques, we have used two independent measures, both well accepted in the literature. The first, *vertex classification*, first introduced by Girvan and Newman [3], has become a popular measure for the evaluation of community identification algorithms:

> *The criterion for deciding correct classification is as follows. We find the largest set of vertices that are grouped together by the algorithm in each of the four known communities. If the algorithm puts two or more of these sets in the same group, then all vertices in those sets are considered incorrectly classified. Otherwise, they are considered correctly classified. All other vertices not in the largest sets are considered incorrectly classified.* [17]

Danon et. al. [18] propose a more rigorous approach, in the use of *normalized mutual information* (NMI) [19]. NMI may be used as a measure of the similarity between the real communities, and the discovered communities, in a graph. If we partition the vertices of a graph with N vertices into $\Omega = \{\omega_1, \omega_2, ..., \omega_K\}$, where ω_k is the set of vertices contained in the real community k, and $\mathbb{C} = \{c_1, c_2, ..., c_J\}$, where c_j is the set of vertices contained in the discovered community j, we may calculate the mutual information by:

$$I(\Omega, \mathbb{C}) = \sum_k \sum_j \frac{|\omega_k \cap c_j|}{N} \log_2 \frac{N|\omega_k \cap c_j|}{|\omega_k||c_j|} \tag{2}$$

and the entropy of each partition by:

$$H(\Omega) = -\sum_k \frac{|\omega_k|}{N} \log_2 \frac{|\omega_k|}{N} \tag{3}$$

Normalized mutual information is then calculated as follows:

$$NMI(\Omega, \mathbb{C}) = \frac{I(\Omega, \mathbb{C})}{[H(\Omega) + H(\mathbb{C})]/2} \tag{4}$$

III. Path-sharing

The aim of the path-sharing measure is to find cluster boundaries, and hence to identify clusters, in graphs. Put simply, the path-sharing between a pair of vertices, i and j, is the ratio of edges which exist between the subgraph formed by vertex i and its neighbours, and the subgraph formed by vertex j and its neighbours, to all such possible edges. This ratio includes the edge between vertices i and j, hence a path-sharing value of 0 is only possible when i and j are not connected to each other. More formally, for an undirected graph $G = (V, E)$, the path-sharing for an edge $(i, j, w) \in E$, between vertices $i, j \in V$, with weight w, is calculated as follows:

- Let V_i denote the set consisting of vertex i and its neighbours, not including vertex j, i.e. $V_i = \{i\} \cup \{u | u \in V, u \neq j, (i, u, w) \in E\}$.
- Similarly, let $V_j = \{j\} \cup \{u | u \in V, u \neq i, (j, u, w) \in E\}$.
- Let $V_{i \cap j}$ denote the set consisting of those vertices which are neighbours of both i and j, i.e. $V_{i \cap j} = V_i \cap V_j$.
- Let $E_{V_i V_j}$ denote the set of edges which lie between vertices in V_i and V_j, i.e. $E_{V_i V_j} = \{(u, v, w) | u \in V_i, v \in V_j, (u, v, w) \in E\}$.
- Let $W_{V_i V_j}$ denote the sum of all weight values corresponding to those edges in $E_{V_i V_j}$, i.e. $W_{V_i V_j} = \sum w | (u, v, w) \in E_{V_i V_j}$
- Now, the path-sharing value for the edge which lies between vertices i and j is given as the ratio of the sum of the weights of the edges which exist between vertices V_i and V_j to its maximum possible value[1]:

$$P_{ij} = \frac{W_{V_i V_j}}{|V_i|.|V_j| - |V_{i \cap j}|} \tag{5}$$

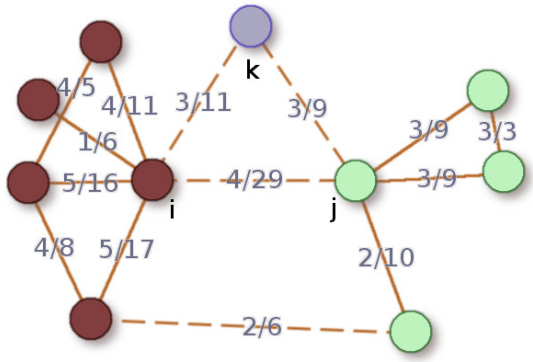


Fig. 1: Path-sharing example.

Consider Figure 1, displaying an unweighted graph (i.e. all edge weights are equal to 1.0) containing two vertices, i and j, and their respective neighbours. Using the notation expressed above, the red (darker) vertices, that is, vertex i and its neighbours, form the set V_i, whereas the green (lighter) vertices (vertex j and its neighbours) form the set V_j. Vertex k is a member of both V_i and V_j, hence $V_{i \cap j} = \{k\}$. The path-sharing P_{ij} for the edge which lies between vertices i and j is calculated simply by summing the weights of all edges which lie between the sets V_i and V_j, that is, the dashed lines in Figure 1. This sum is then divided by the total weight of

[1]The edge weights are assumed to lie between 0.0 and 1.0.

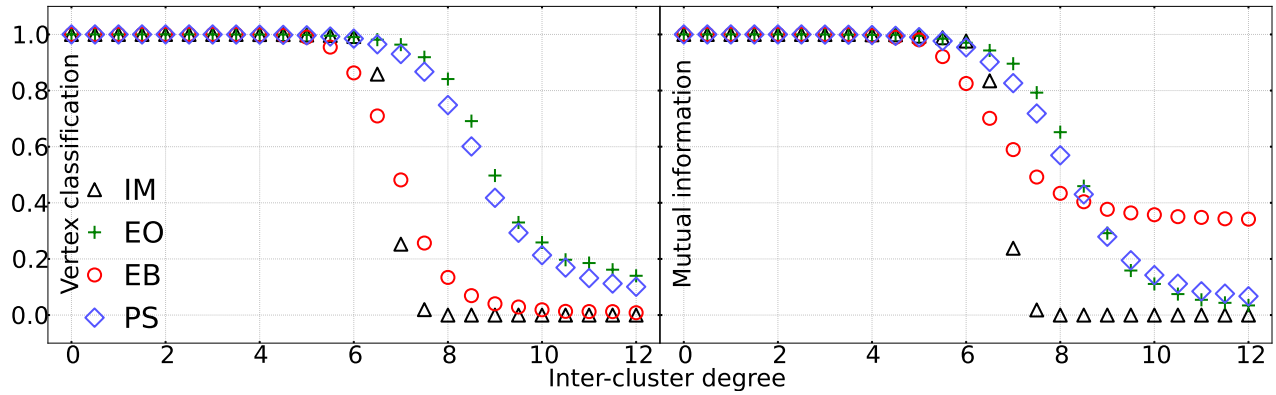


Fig. 2: Results for Girvan and Newman's artificially clustered networks. Each point represents an average over 100 randomly generated graphs. *PS* refers to path-sharing, *EB* to edge-betweenness, *EO* to extremal optimization, and *IM* to Infomap. Standard error for each point is less than 0.05, so is not shown.

all such possible edges. In this example, the summed weight of all edges which lie between V_i and V_j is 4, and the total possible weight is $(6 \times 5) - 1$ (1 is subtracted to account for vertex k), giving $P_{ij} = \frac{4}{29} \approx 0.138$.

The computational complexity of path-sharing is proportional to the number of edges m, and the average degree $\langle k \rangle$, of the graph. For every edge in the graph, the weights of other edges which exist between the subgraphs formed at the two endpoints of the edge (V_i and V_j in Equation 5) must be summed. On average, the size of such a subgraph will be $\langle k \rangle$; every pair of vertices between the two subgraphs must be checked, meaning that the complexity for a single edge is $O(\langle k \rangle^2)$. This is a worst-case scenario, as there will usually be overlap between the two subgraphs ($V_{i \cap j}$ in Equation 5). Performing this for every edge gives an overall worst-case complexity of $O(m\langle k \rangle^2)$. When path-sharing is used within a divisive clustering algorithm, substantial cost savings can be achieved; this is due to the fact that Equation 5 requires only a local calculation upon portions of a graph, in a similar vein to the edge-clustering coefficient, introduced by Radicchi et. al. [20]. This is in contrast to edge-betweenness, which requires calculation over an entire component to produce a value for a single edge. After an edge between two vertices i and j has been removed from the graph, path-sharing values need only be re-calculated for any remaining edges which are adjacent to vertices i or j, or which lie between the two sets V_i and V_j.

IV. Edge-betweenness

Shortest-path edge-betweenness (or simply edge-betweenness) is the de-facto standard betweenness measure for community identification in graphs. It is described in detail by Newman and Girvan [4], building upon foundational work by Anthonisse [21] and Freeman [22]. The aim of the edge-betweenness measure is to determine the relative importance of edges in a graph, by counting the number of shortest paths, or *geodesics*, between pairs of vertices, which pass through each edge.

A high betweenness score on an edge indicates that the edge plays a relatively important role in allowing vertices upon either side to communicate; this indicates that the edge lies on a boundary between more densely connected clusters. A low betweenness score, on the other hand, indicates that a number of alternative pathways to that edge exist in the graph.

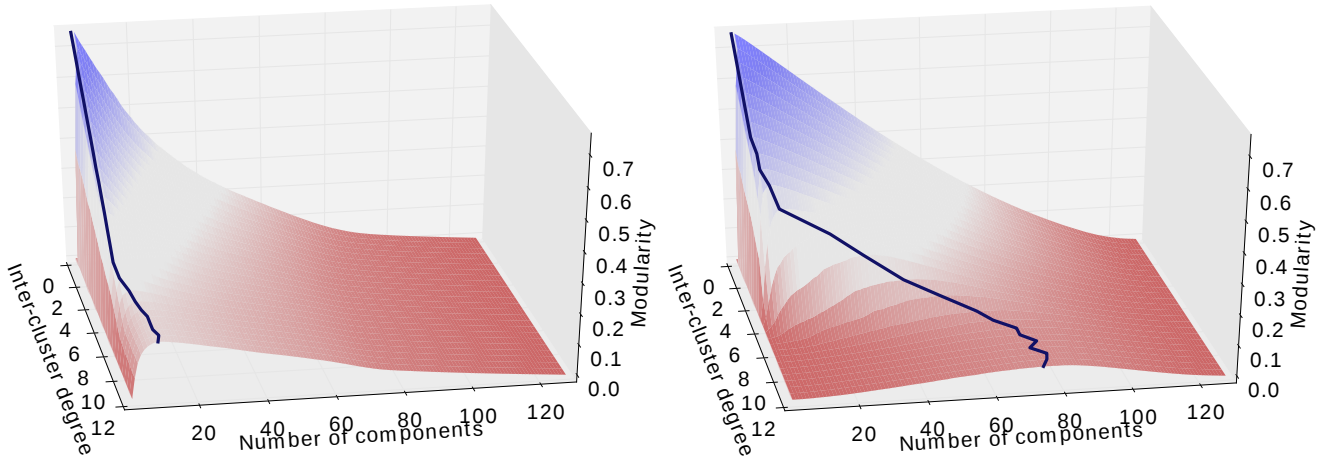
Consider an undirected, unweighted graph with n vertices and m edges. Starting with one *source* vertex v, the number of geodesics, w_i from vertex v to all other vertices i in the graph are counted; this is achieved by a breadth-first traversal, with complexity $O(m)$. The length of each geodesic d_i, from vertex v to all other vertices i, may also be calculated in the same traversal. Once the geodesic counts and lengths have been calculated, we may begin the process of iteratively assigning betweenness values to every edge in the graph, starting with those edges that are farthest from vertex v.

The betweenness value for an edge e_{ij}, lying between two vertices i and j, where d_j is greater than d_i, is calculated as 1 plus the sum of the betweenness scores over all edges e_{jk}, where d_k is greater than d_j. This sum is then multiplied by the ratio w_i/w_j. Terminating, or leaf edges, are simply assigned a value of w_i/w_j. Any edges e_{ab}, where $d_a = d_b$, are assigned a value of 0.

This process is repeated, using every other vertex as the source vertex, giving an overall complexity of $O(2mn)$. The betweenness values upon each edge are tallied to give a final edge-betweenness score.

V. Benchmark networks

A standard benchmark for evaluating community identification algorithms is the use of artificial clustered networks, as proposed by Girvan and Newman [3]. Edges are placed between pairs of vertices at random; intra-cluster edges (edges between vertices in the same cluster) are included with probability P_{in}, and inter-cluster edges included with probability P_{out}. We generated a large number of these graphs, each containing 4 clusters of 32 vertices each. P_{in} and P_{out} were varied such that the overall average degree remained



(a) Path-sharing

(b) Edge-betweenness

Fig. 3: Modularity response as the divisive clustering algorithm progresses; the solid blue lines show the maximum modularity value reached, as it varies with the inter-cluster degree.

constant at 16. Community identification was applied to each graph, with path-sharing giving similar results to extremal optimization, as shown in Figure 2.

NMI appears to respond in a rather lenient manner towards edge-betweenness, asymptotically approaching ~ 0.35 as the community structure weakens. This is due to the fact that edge-betweenness has a strong tendency to disconnect a large portion of the vertices in graphs with weaker community structure. The other techniques do not exhibit this behaviour. Figure 3 explores this further, by displaying the change in modularity as the divisive algorithm progresses, and the graph splits into more and more components. With path-sharing, the maximum modularity value is reached quickly, when only a small number of components have formed. In contrast, as the community structure weakens, the maximum modularity value reached by edge-betweenness tends to be found only after the graph has broken into a large number of components[2].

Lancichinetti et. al. [5], [23] argue that Girvan and Newman's artificially clustered networks are too artificial, in that they are not representative of real-world networks. Such 'real-world' networks are characterised by a heterogeneous degree distribution, which may be modelled with a power law. The authors also suggest that community sizes in real-world networks are not homogenous, and can, again, be modelled with a power law. To this end, they propose a new benchmark for evaluation of community identification algorithms, a set of randomly generated networks in which both vertex degree and community size are defined by independent power law distributions. The algorithm for creating such a graph takes as its input: the number of vertices N; average degree $\langle k \rangle$; two power law exponents, γ and β, which define the distribution for degree and community size respectively; and finally the *mixing parameter* μ, which defines the approximate proportion of inter- and intra-cluster edges for each vertex. Values of μ less than, or greater than 0.5, will generate graphs with a stronger, or weaker community structure, respectively.

We again generated a large number of these graphs. For compatibility with Lancichinetti and Fortunato's comprehensive comparison of community identification techniques [16], we tested two classes of networks, each containing 1000 vertices, with the average degree $\langle k \rangle = 20$ and maximum degree 50. Degree and community size distributions were defined with $\gamma = 2.0$, and $\beta = 1.0$. Community sizes were limited to two ranges, $(10, 50)$, and $(20, 100)$. The value of μ was used to vary the strength of the community structure, from 0.0 to 0.8. Results are shown in Figure 4[3].

Path-sharing and Infomap give similar performance on these networks, with the difference that Infomap degrades quite quickly as the community structure weakens. In contrast, path-sharing degrades more gracefully as μ is increased. This is not necessarily an advantage: the all-or-nothing response given by Infomap may be a more desirable characteristic if the aim is simply to test the presence of community structure in a network. However, path-sharing may be a more appropriate technique if the goal is the classification of individual vertices into communities.

VI. Image Segmentation

Graph theory is a common approach to the problem of segmenting an image into its constituent parts. The image is modelled as a graph, with edges between pixels weighted by their spatial proximity, and intensity similarity. Shi and Malik [11] present a simple approach to graph creation, requiring three parameters: σ_F, the intensity scaling parameter; σ_X, the

[2]It is not possible to display the same data for extremal optimization or Infomap, due to algorithmic differences.

[3]Results for edge-betweenness reproduced with permission from [16]. For these results, Lancichinetti et. al. used a variant of normalized mutual information, updated to support overlapping communities [24]. This variant, whilst not giving exactly the same results as Danon et. al.'s original definition, gives results which are close enough to make this comparison meaningful.

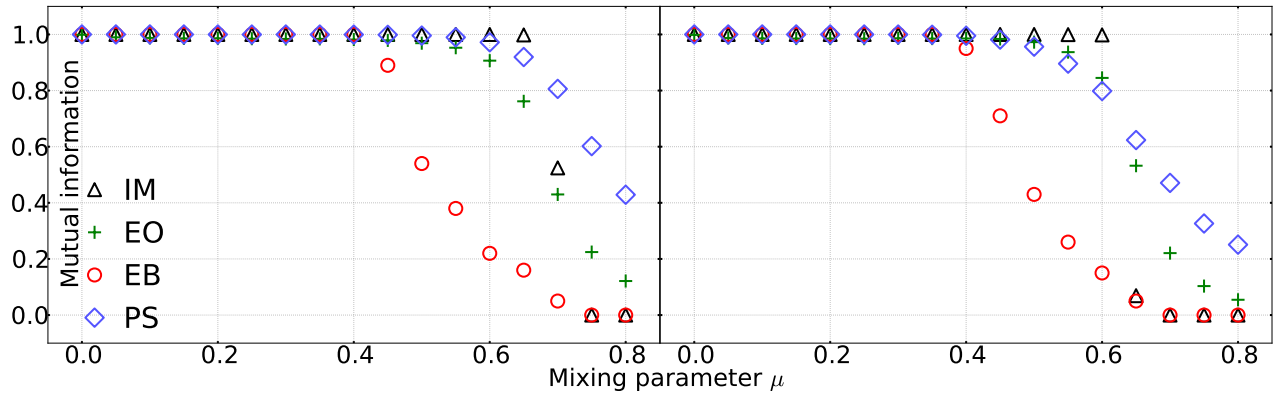


Fig. 4: Results for Lancichinetti et. al's artificially clustered networks. Each point represents an average over 100 randomly generated graphs. Results for community size range $(10, 50)$ are on the left, and for range $(20, 100)$ on the right. Standard error for each point is less than 0.05, so is not shown.

distance scaling parameter; and r, the cutoff distance. Every pixel in the image is added as a vertex in an undirected graph $G = (V, E)$; edge weights w_{ij} between each pair of pixels $i, j \in V$ are defined by:

$$w_{ij} = e^{\frac{-|F(i)-F(j)|^2}{\sigma_F^2}} \times \begin{cases} e^{\frac{-|X(i)-X(j)|^2}{\sigma_X^2}} & \text{if } |X(i) - X(j)| < r \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $|F(i) - F(j)|$ is the difference in intensity between pixels i and j[4], and $|X(i) - X(j)|$ is the distance between pixels i and j.

We created graphs in this fashion for two test images, obtained from the Berkeley Segmentation Dataset [25]. The test images and results are shown in Figure 5. Path-sharing succeeded in segmenting and identifying the major features of each test image. A major limiting factor in this approach is the need to adjust the graph creation parameters for individual images, to ensure that image features are well represented in the graph model.

VII. Conclusion

In summary, we have presented path-sharing, a new measure of betweenness for use in community identification, and demonstrated that it gives good performance, when compared with high performance techniques such as extremal optimisation and Infomap, upon artificial benchmark networks. We also demonstrated the practical applications of path-sharing by applying it to image segmentation. It is interesting to note that a simple, deterministic measure, such as path-sharing, is able to give results similar to that of high performance optimisation techniques. Path-sharing provides a fundamental measure of the connectivity between two groups of vertices in a network, and it is our hope is that the research community will find uses for it in areas beyond community identification.

[4]We have dealt only with greyscale images, where pixel values have been scaled to lie between 0 and 255.

Acknowledgements

The author would like to thank Dr. Lubica Benuskova for feedback regarding the manuscript, Dr. Mark Newman for making a large collection of network data publicly available on his website, at http://www-personal.umich.edu/~mejn/netdata/, Dr. Santo Fortunato, for providing code to generate community identification benchmark graphs [5] online at http://sites.google.com/site/santofortunato/, Dr. Sergio Gómez, for providing an implementation of extremal optimzation online at http://deim.urv.cat/~sgomez/index.php, and Dr. Martin Rosvall for providing an implementation of Infomap online at http://www.tp.umu.se/~rosvall/code.html. This research was funded by a University of Otago PhD Scholarship. All software used for generating and evaluating graphs is available online at http://github.com/pauldmccarthy/.

References

[1] B. W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," *Bell System Technical Journal*, vol. 49, no. 2, pp. 291–307, 1970.

[2] J. Scott, *Social Network Analysis: A Handbook*, 2nd ed. Sage Publications, 2000.

[3] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[4] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, p. 026113, 2004.

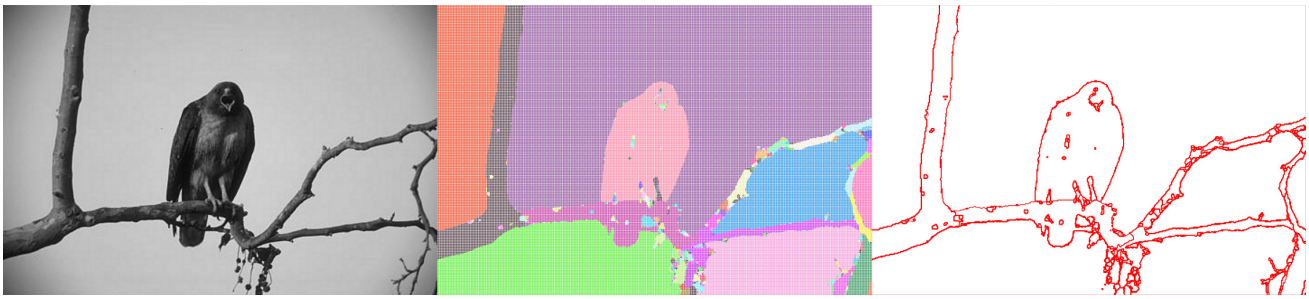
[5] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, p. 046110, 2008.

[6] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical Review E*, vol. 72, no. 2, p. 027104, 2005.

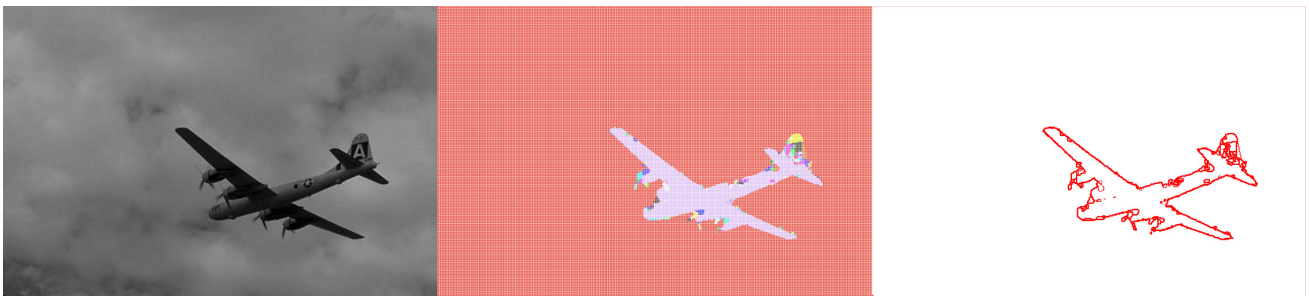
[7] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[8] C. T. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Transactions on Computers*, vol. C-20, no. 1, pp. 68–86, 1971.

[9] Z. Wu and R. Leahy, "An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1101–1113, 1993.



(a) Bird image ($\sigma_F = 25, \sigma_X = 6, r = 1.5$)



(b) Plane image ($\sigma_F = 25, \sigma_X = 6, r = 1.5$)

Fig. 5: Results of applying path-sharing to an image segmentation problem. Raw test images are shown on the left; Graph vertices after community identification was applied are shown in the centre, coloured according to their enclosing component. Boundaries between identified communities are shown on the right.

[10] L. Grady, "Random Walks for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[11] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[12] J. Yoon, A. Blumer, and K. Lee, "An algorithm for modularity analysis of directed and weighted biological networks based on edge-betweenness centrality." *Bioinformatics*, vol. 22, no. 24, pp. 3106–8, 2006.

[13] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Second Edition ed., ser. Springer Series in Statistics. Springer, 2009.

[14] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

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

[16] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Physical Review E*, vol. 80, no. 5, p. 056117, 2009.

[17] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, p. 066133, 2004.

[18] L. Danon, J. Dutch, A. Diaz-Guilera, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 9, p. 09008, 2005.

[19] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[20] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences*, vol. 101, no. 9, pp. 2658–2663, 2004.

[21] J. M. Anthonisse, "The Rush In A Directed Graph," Stichting Mathematisch Centrum, Amsterdam, Tech. Rep., 1971.

[22] L. C. Freeman, "Centrality in Social Networks: Conceptual Clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1979.

[23] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Physical Review E*, vol. 80, no. 1, p. 016118, 2009.

[24] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, p. 033015, 2009.

[25] D. R. Martin, C. Fowlkes, D. Tal, and J. Malik, "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics," in *Proceedings of the 8th International Conference on Computer Vision*, 2001, pp. 416–423.