

# Estimating Network Parameters for Selecting Community Detection Algorithms

LETO PEEL

This paper considers the problem of algorithm selection for community detection. The aim of community detection is to identify sets of nodes in a network which are more interconnected relative to their connectivity to the rest of the network. A large number of algorithms have been developed to tackle this problem, but as with any machine learning task there is no “one-size-fits-all” and each algorithm excels in a specific part of the problem space. This paper examines the performance of algorithms developed for weighted networks against those using unweighted networks for different parts of the problem space (parameterised by the intra/inter community links). It is then demonstrated how the choice of algorithm (weighted/unweighted) can be made based only on the observed network.

Manuscript received October 26, 2010; revised May 05, 2011; released for publication June 11, 2011.

Refereeing of this contribution was handled by Mujdat Cetin.

Author’s address: Advanced Technology Centre, BAE Systems, Bristol, UK, E-mail (letto.peel@baesystems.com).

---

1557-6418/11/\$17.00 © 2011 JAIF

## 1. INTRODUCTION

The study of large scale networks has revealed a number of properties about the behaviour and topology of naturally occurring networks. One such property is the presence of community structures; sets of nodes in a network which are more interconnected relative to their connections to the rest of the network. The aim of community detection is to identify these structures. Community detection is a problem which has attracted much interest in recent years [7], [12], [19], [23], [27] and has consequently produced a wide range of approaches to the problem; an in-depth review of most contemporary methods is given in [8].

One of the reasons why the ability to detect communities is so attractive lies in the phenomenon known as *assortative mixing*, where entities in a network are observed to associate preferentially with similar entities. This suggests that detecting communities may be used for identifying entities which share common attributes or purposes. An example of community structures corresponding to entity similarity is given in [20] where community structures in a friendship network correspond to similarities in race and age. The wide range of complex systems that can naturally be expressed as networks (human interaction patterns, metabolic networks, WWW, and the brain) implies that community detection has applications spanning domains as diverse as biology [11], [14], [29], sociology [2], [11], [28], computer science [26], [30] and intelligence [1], [10], [16].

The implications of community detection in the intelligence domain are that it could be used to identify groups of people who share common goals or purposes. To this effect, community detection could potentially be used to constrain the inference problem when investigating or detecting malicious activities, e.g. rather than monitoring all people, use community detection as a pre-processing step to select a subset of people to monitor. In this setting, the network nodes would represent people and the links would represent interactions or relationships between them; such a network can be constructed from a database of phone records, email logs or other transactional data.

With a large selection of algorithms available to undertake the task of community detection, choosing an appropriate algorithm becomes problematic. This is largely due to the lack of formal or commonly accepted evaluation procedures. The networks used to evaluate community detection tend to be a small selection of real networks and/or networks generated from simple models, where these networks vary widely between authors. Recent work to address this has focused on developing benchmark networks [17] on which comparative analysis [18] can be drawn to determine the reliability of different algorithms. However, it is commonly accepted across the machine learning community that there is no *one-size-fits-all* solution and so this work considers the idea that for different situations, different classes

of algorithms may outperform other classes of algorithms. The range of community detection algorithms in itself poses the intelligence analyst with the challenge of choosing an appropriate solution or combinations of solution techniques for the specific problem at hand. It is therefore desirable to be able to provide the intelligence analyst, who will likely not have expert knowledge of these algorithms, with appropriate guidance. This paper considers the problem of automatically selecting community detection algorithms based on observations of the community structure.

The algorithm selection problem can be solved in part by design. For example, consider the class of algorithms involving modularity [22] optimisation. Simulated annealing [15] produces good results, but for large networks it may be preferable to substitute performance for speed by using a greedy algorithm such as in [5]. In this situation the trade off is straight forward, a choice of speed vs. quality. However, some algorithms address other limitations for which the trade off is not so clear cut. For example the modularity function suffers from a resolution limit [9] meaning that in some networks several communities become merged into one (e.g. communities consisting of two cliques connected by a single link). Some algorithms [3], [4] address this issue but it is not clear what weaknesses (if any) this might introduce. In cases such as these further investigation is required to determine the appropriate algorithm. In this work the class of algorithms which incorporate link weight information is examined to determine when these are most appropriate.

It has been previously observed how structural properties of communities affect the performance of community detection algorithms [18]. These properties cannot be measured from the network data alone as they require knowledge of the underlying community assignment. The main contribution of this work is to demonstrate how these structural properties can be estimated from features of the observed network. Therefore a prediction about which algorithm will perform best can be made. This is achieved by considering algorithms for weighted networks and algorithms for unweighted networks as two separate classes and demonstrating how the performance of these two classes differs across the problem space (defined in Section 2). Finally, a Support Vector Machine (SVM) [6] is used to classify the networks according to the algorithm which will perform best.

The rest of the paper is organised as follows: Section 2 defines the problem space by defining the network and community structure types and the target algorithm classes. The performance of the algorithm classes with respect to the structural parameters is evaluated in Section 3. Section 4 describes the *observable* network parameters and how a mapping can be made from these to the underlying structural parameters. The results of

using the observable parameters to choose an appropriate class of algorithms are given in Section 5. Conclusions are given in Section 6.

## 2. PROBLEM SPACE

A network is a structure made up of nodes, representing entities, and links or edges, representing relationships or interactions between entities. The total number of links connected to a node is known as its degree. The network links may also have weights associated with them which may represent the relative importance of the link. For example, in an interaction network representing a phone record database, the nodes would represent people and the links phone calls. The link weights could then represent the frequency of calls. Network links may also be directed, but this will not be considered in this work.

The premise of community detection is that there is some underlying assignment of nodes to communities which has to be discovered. But despite the large amount of literature on the subject there is still a lack of agreement on what defines a community beyond the intuitive concept that community structures have more intra-community links than inter-community links. Without a common definition it is difficult to draw a comparison between algorithms. However, it may not be necessary (or even desirable) to define a specific common definition of community, as definitions may be dependent on the application. Instead, perhaps all that is required is a suitably comprehensive parameter set for describing the space of community types and structures of interest.

A reasonable starting point is the parameter set used to generate networks and communities using the Lancichinetti-Fortunato-Radicchi (LFR) benchmark generator [17] as not only do these describe a number of network properties, but by using the generator it is possible to obtain networks and community assignments with those properties. This parameter set is described in Section 2.1.

### 2.1. Network-Community Parameterisation

The parameter set used to describe the problem space are the parameters used by the LFR benchmark which is fully described in [17]. The LFR benchmark was designed to generate datasets to test community detection algorithms and mimic the observed properties of large-scale real complex networks [21], such as power-law degree and community distribution.

The parameters are best described in the context of the graph generation procedure:

1.  $N$  nodes are assigned to communities such that the community size distribution conforms to a power-law with minus exponent  $\tau_2$ .
2. Each node is assigned a degree such that the degree distribution conforms to a power law with minus exponent  $\tau_1$  and mean degree  $k$ .

3. Links are initially assigned randomly according to the degree distribution. A topological mixing parameter,  $\mu_t$ , is set to define the proportion of each node's links which link outside its community. Topological consistency with this parameter is achieved through an iterative re-wiring procedure.

4. Each node is then assigned a strength according to a power-law distribution with minus exponent  $\beta$ . The strength of a node is the weighted analogy of degree and as such represents the sum of the weights of the links for a given node.

5. To assign the link weights a similar process to step 3 is carried out according to the weight mixing parameter,  $\mu_w$ .

Networks and their communities can be generated for a specified set of parameter values using the above procedure. In addition other constraining parameters can be specified including the maximum degree ( $\max k$ ), and the minimum and maximum sizes of communities ( $c_{\min}$  and  $c_{\max}$  respectively). Using these parameters to describe the problem space has the advantage that by using the generative procedure outlined above it is possible to obtain a network for a given set of parameter values. It is accepted that these may not be a full set of parameters to comprehensively describe the space of all possible network-community structures. Even so, the space is one of high dimensionality and so full exploration of all the parameters is beyond the scope of this paper and remains for future work. To constrain the problem, the values of all parameters were fixed with the exception of  $\mu_t$  and  $\mu_w$ , which from initial tests were found to have the greatest impact on use of link weights.

## 2.2. Algorithm Overview

The algorithm selection problem has been constrained to choosing between the class of algorithms which use link weight information and the class that does not. In light of this, it was decided to use algorithms suitable for unweighted or weighted networks. This way a controlled comparison can be drawn between the performances of the unweighted and weighted algorithms without being concerned about differences in algorithms. Two such algorithms are examined:

- *Infomap* [27]: This algorithm approaches the community detection problem by identifying a duality between community detection and information compression. By using random walks to analyse the information flow through a network it identifies communities as modules through which information flows quickly and easily. Coding theory is used to compress the data stream describing the random walks by assigning frequently visited nodes a shorter codeword. This is further optimised by assigning unique code-words to network modules and reusing short code-words for network nodes such that node names are

unique given the context of the module. This two level description of the path allows a more efficient compression by capitalising on the fact that a random walker spends more time within a community than moving between communities.

- *COPRA* [13]: This is an extension of the label propagation based RAK algorithm [25]. The algorithm works as follows; to start, all nodes are initialised with a unique label. These labels are then updated iteratively, where a node's new label is assigned according to the label used most by its neighbours. If there is more than one most frequently occurring label amongst the neighbours, then the label is chosen randomly. At termination of the algorithm, nodes with the same label are assigned to the same community. The **Community Overlap P**ropagation **A**lgorithm (**COPRA**) extends the RAK algorithm to deal with the possibility of overlapping communities (although this aspect of community detection is not explored within this work). This is done by augmenting the label with a belonging factor such that for a given node these sum to 1. To prevent all nodes becoming a member of all communities, a threshold is set below which the labels are discarded. Due to the stochastic nature of the algorithm, the algorithm is run a number of times and the "best" community assignment is decided according to the one which has the highest modularity [22]. In the weighted instance of the algorithm, the weights of the network are incorporated by weighting the frequency of the labels according to the link weight connecting the respective node.

## 3. ALGORITHM PERFORMANCE

A number of different metrics are used in the literature to measure the performance of community detection algorithms, however the Normalised Mutual Information [19] metric is one which has become fairly standard recently and so will be used here. This metric provides a measure of similarity between the algorithm output assignment and the true community assignment, where a value of 1 denotes a perfect match. Using the networks described in Section 2.1, experiments were run to examine the effect of varying the two mixing parameters  $\mu_t$  and  $\mu_w$ , the results of which can be seen in Fig. 1.

Fig. 1 shows the mutual information scores for the weighted algorithms (COPRAw, INFOMAPw) and unweighted algorithms (COPRAuw, INFOMAPuw) as  $\mu_w$  is changed. The plots (a)–(d) show the performance for different values of  $\mu_t$ . Each point on the graphs represents the average mutual information over 100 generated networks (using the LFR benchmark generator described in Section 2.1) with the indicated parameter values. It can be seen that the unweighted algorithms perform well when  $\mu_t$  is low and are unaffected by  $\mu_w$  for all values  $\mu_t$ . This is because these algorithms only

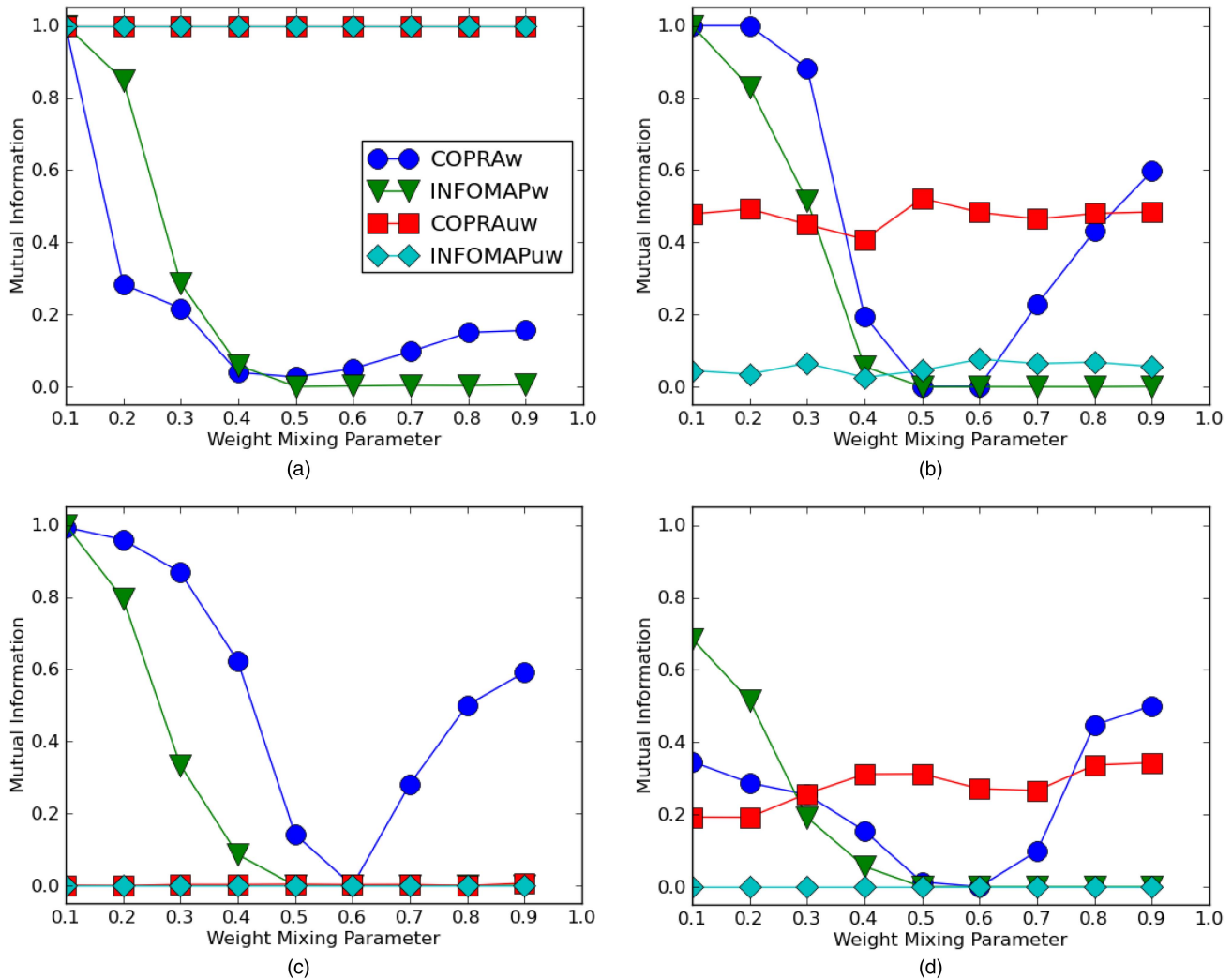


Fig. 1. Mutual information scores (y-axis) as  $\mu_w$  (x-axis) changes. Each subplot shows a different fixed value for  $\mu_t$ . The values of the other parameters were fixed:  $N = 100$ ,  $k = 25$ ,  $\max k = 50$ ,  $\tau_1 = 2$ ,  $\tau_2 = 1$ ,  $\beta = 1.5$ ,  $c_{\min} = 15$ ,  $c_{\max} = 50$ . (a)  $\mu_t = 0.2$ . (b)  $\mu_t = 0.4$ . (c)  $\mu_t = 0.6$ . (d)  $\mu_t = 0.8$ .

rely on the topological information. The weighted algorithms on the other hand are affected by both parameters, but are seen to consistently perform well for low  $\mu_w$ . The effect of  $\mu_t$  is probably best observed in Fig. 2. Here it can be seen that the weighted algorithms perform well when  $\mu_t$  is at least as high as  $\mu_w$  (in this case  $\mu_w = 0.3$ ). A similar observation was made in [18] where it was seen that weighted algorithms performed better overall at  $\mu_t$  values of 0.5, in comparison to lower values. It was explained that the reason for this is that a low  $\mu_t$  relative to  $\mu_w$  means that there is a lower proportion of inter-community links relative to the proportion of inter-community weights. The effect of this is that a small number of inter-community links receive high link weights relative to the intra-community weights, see Fig. 4.

The effect of this is that there are regions of the problem space, parameterised by community mixing proportions, in which a weighted algorithm will outperform an unweighted one and vice versa. This can be seen in

Fig. 2 where the two regions are labelled w (weighted) and uw (unweighted). This result indicates that a choice can be made, based on the community structure, as to the class of community detection algorithms.

In order to take advantage of this information and select the best class of algorithms for a given network, some knowledge of the underlying community structure is required. It may be possible to make some assumption about the communities that are sought after based on some knowledge of the specific domain. In most community detection problems however, this information about the community structure is unknown.

#### 4. PARAMETER ESTIMATION

In order to use the information from the previous section, it is required to know the values of the mixing parameters of the communities. Without knowledge of the communities (i.e. prior to community detection) it is not possible to evaluate these parameters. In this section it will be shown how parameters of the observable

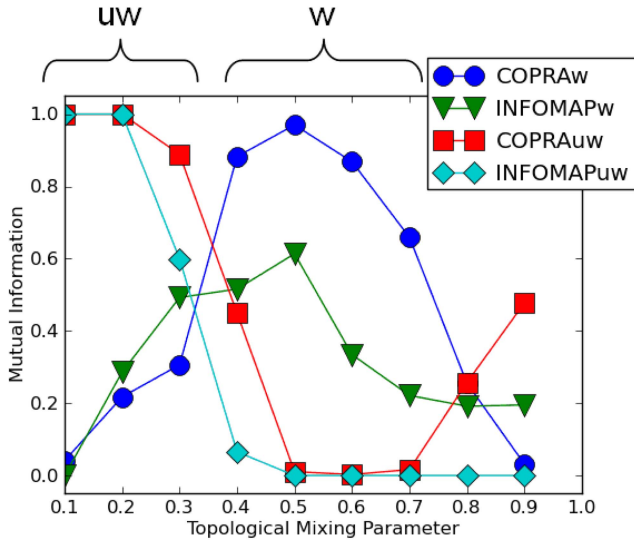


Fig. 2. Mutual information scores for the weighted (w) and unweighted (uw) algorithms as  $\mu_t$  is varied. The value of  $\mu_w$  is fixed at 0.3. It is noticeable that the two classes of algorithm perform for complimentary settings of  $\mu_t$ .

network can be mapped to these community parameters and how these values can be used to build a classifier to determine the class of community detection most suitable for the given network.

#### 4.1. Observable Parameters

There are a range of metrics associated with describing network topology: degree distribution, average diameter, and centrality measures are a few of them. The problem here is that a parameter is required which describes the way that the community structures interact, without explicitly knowing the community structures.

To approach this, the node measure called clustering coefficient [31] is considered. This is defined as:

$$C_{\text{unweighted}}^{(v)} = \frac{\sum_{i,j \in N_v} e_{ij}}{k_v(k_v - 1)/2} \quad (1)$$

where the local clustering coefficient,  $C_{\text{unweighted}}^{(v)}$ , represents the proportion of the neighbours,  $N_v$ , of node  $v$  which are connected (i.e. edge  $e_{ij} = 1$  if there is a link between neighbouring nodes  $i$  and  $j$ ) out of the possible connections between its neighbours,  $k_v(k_v - 1)/2$ .<sup>1</sup> It was found that the mean value of the local clustering coefficient, taken over all the nodes in the network, showed a strong correlation with the topological mixing parameter,  $\mu_t$  (Fig. 3(a)). This is similar to an observation in [24] where high clustering coefficients were observed in Girvan-Newman networks [11] with low ratios of inter- to intra-modular connectivity (a parameter

<sup>1</sup>If the degree of a node is less than or equal to 1 then the clustering coefficient is defined as 0.

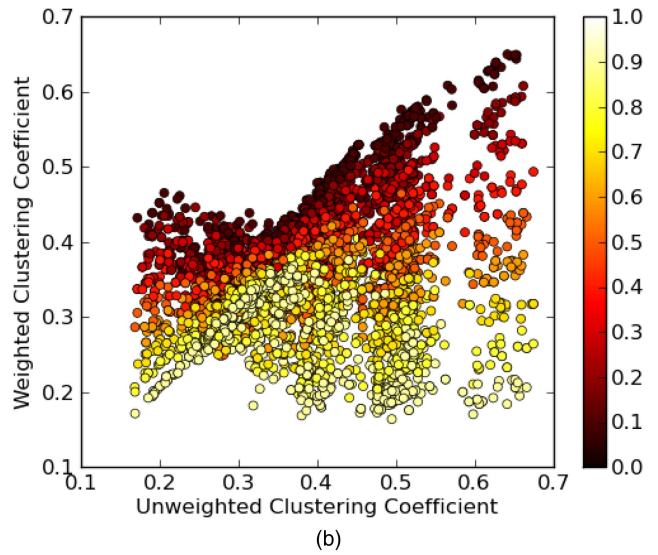
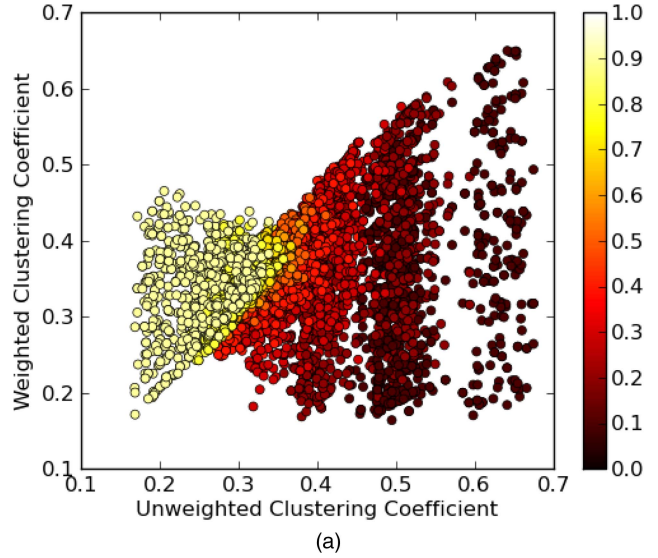


Fig. 3. Scatter plots of the unweighted (x-axis) and weighted (y-axis) mean local clustering coefficient for the networks in Fig. 1, (a) shows the value of the topology mixing parameter,  $\mu_t$ . Similarly (b) shows the weight mixing parameter,  $\mu_w$ .

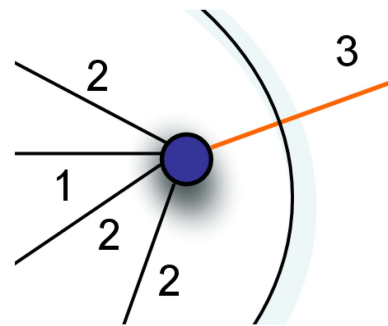


Fig. 4. An example node with links and weights from a network with  $\mu_t = 0.2$  and  $\mu_w = 0.3$ . As a result the single inter-community link (orange) receives a higher weight relative to the intra-community links.

which bears resemblance to the topological mixing parameter,  $\mu_t$ , in the more realistic LFR networks used here).



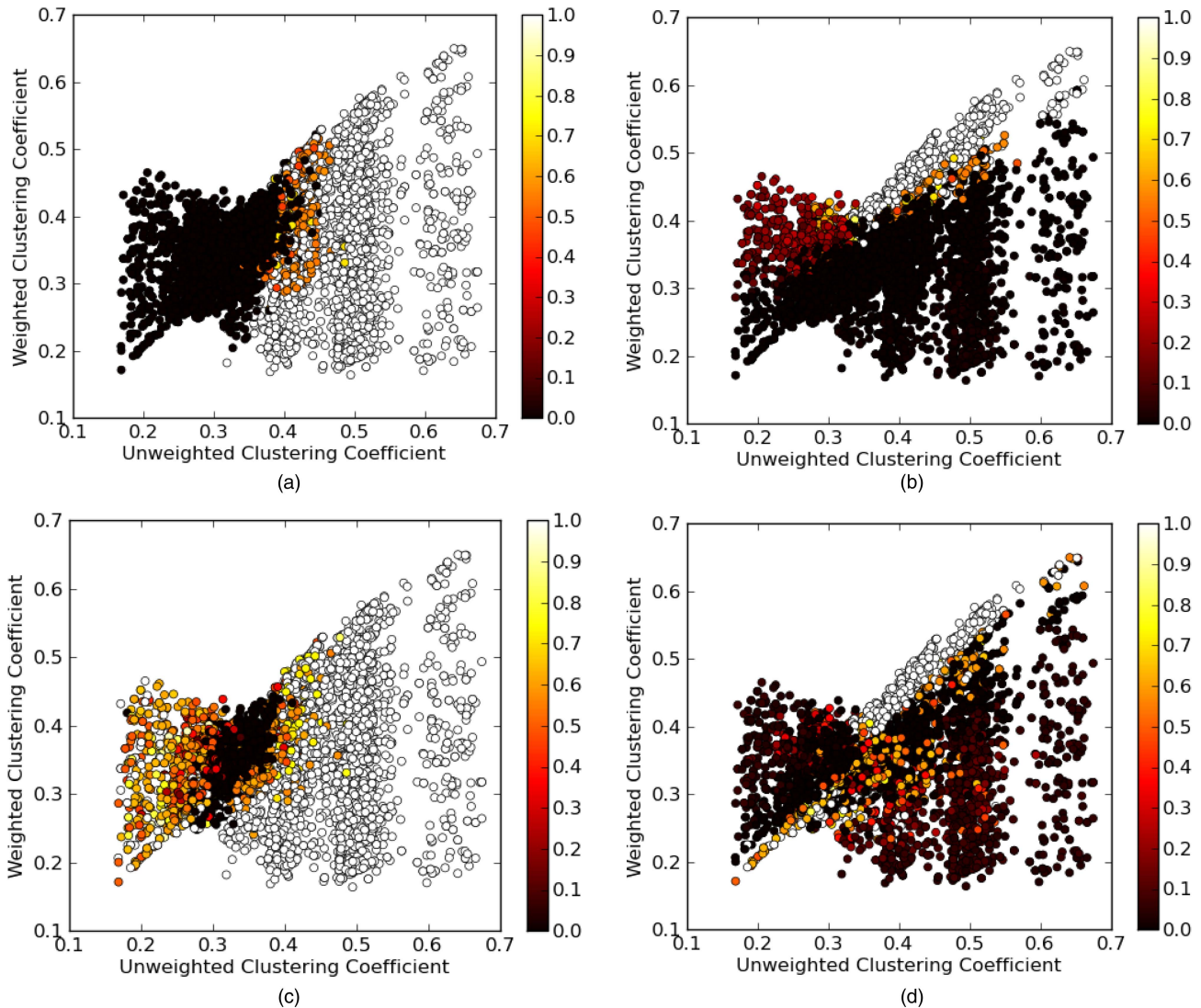


Fig. 5. Clustering coefficients scatter plots (the same networks used in Fig. 1) with colours showing the mutual information score for (a) unweighted infomap, (b) weighted infomap, (c) unweighted COPRA and (d) weighted COPRA.

This observation suggests that the mean clustering coefficient could be used to estimate the mixing parameter  $\mu_i$ . If the mean clustering coefficient could be used to estimate the topological mixing then it follows that a weighted extension to this may yield information about the weighted mixing parameter (Equation 2).

$$C_{\text{weighted}}^{(v)} = \frac{\sum_{i,j \in N_v} (w_{vi} + w_{vj}) e_{ij}}{\sum_{i \in N_v} w_{vi} (k_v - 1)} \quad (2)$$

where  $w_{vi}$  is the weight associated with the link between nodes  $v$  and  $i$ . The mean of this value over the network was found to correlate with  $\mu_w$  (Fig. 3(b)). The results in Fig. 3 suggest that the mixing parameters can be estimated from observed network characteristics without knowledge of the community structure.

The reason for this can be explained by considering the general principle of a community; that nodes within a community are more likely to be connected compared to overall probability of connection due to the sparse

nature of the network. Hence, if two neighbours are within the same community, it is reasonable to expect them to be connected. However, if neighbours are not in the same community it is more likely that they are not connected. Based on this reasoning, the local clustering coefficient is an estimate of the individual node's mixing parameter, which averaged over the network yields a global estimate.

#### 4.2. Algorithm Classification using SVM

The results of the previous section suggest that it is possible to estimate the mixing parameters of the communities. Now returning to the reason why it may be useful to estimate these parameters, i.e. to determine the class of algorithm, it is suggested that rather than estimate the mixing parameters and in turn predict the algorithm class, it may be more useful to use the clustering coefficients to directly predict the algorithm class. Fig. 5 shows similar plots as Fig. 4, but with the colour indicating the performance for the different

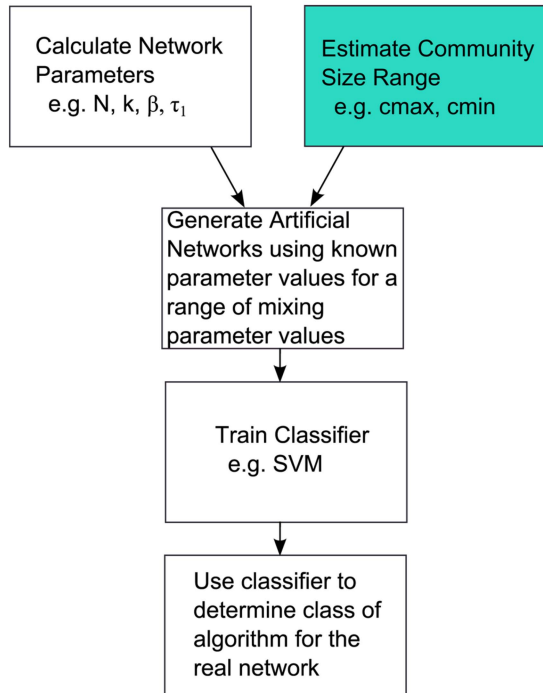


Fig. 6. Algorithm selection pipeline for classification of real networks. The shaded box indicates a step which requires further work to define.

algorithms. It can be seen that the weighted algorithms have a distinctly different performance pattern from the unweighted ones.

In order to confirm that these observable parameters can effectively predict the algorithm class, a simple classifier was built using support vector machines (SVM) [6] with a Gaussian kernel. To do this, each of the networks was assigned a class weighted, unweighted, none based on the class of algorithm which performed best in terms of its mutual information score. A class of “none” was assigned to any network where the mutual information score for the best performing algorithm was below some threshold. The reasoning for this is that for low performance values the output is not meaningful and therefore the choice of algorithm is irrelevant. As SVMs are restricted to two classes, three classifiers were trained (weighted vs. unweighted, weighted vs. none, unweighted vs. none) and the predicted class obtained by using a voting scheme over the three outputs, i.e. the predicted class is the most frequent output of the three classifiers. The results are discussed in Section 5.

#### 4.3. Applicability to Other Networks

So far all the networks examined have had the same network properties except for the mixing parameters. To investigate if this relationship still holds for networks with different parameter settings, the experiments were repeated for: larger networks ( $N = \{1000, 5000\}$ ), small and large community sizes ( $c_{\min} = \{15, 25\}$ ,  $c_{\max} = \{50, 100\}$ ), and different average degree ( $k = \{25, 35\}$ ).

TABLE I  
Classifier Confusion Matrix (100 nodes)

		Predicted Class		
		Weighted	Unweighted	None
True Class	Weighted	160	6	146
	Unweighted	8	369	18
	None	29	51	1213

Fig. 7 shows the clustering coefficient scatter plots indicating the mixing parameter values of the different graphs at each of the parameter settings. It can be seen that as long as the other parameter values are fixed there is still a correlation between the unweighted clustering coefficient and topological mixing parameter  $\mu_t$  and between the weighted clustering coefficient and weight mixing parameter  $\mu_w$ . It can also be seen from the scatter plots that these relationships are more or less the same for most of the different parameter settings. The main differences occur when the community size changes relative to the number of nodes in the network, i.e. as the ratio of community size to network size increases the points on the scatter plots become more compressed.

In order to apply the algorithm classification method presented in this paper to real networks, a classifier could be trained on a set of artificial networks which resemble the real network. This process is outlined in Fig. 6. There are a number of limitations with this method. First, the step which estimates the community size range (indicated by the shaded box) is currently undefined and requires further investigation to identify a method for estimating this range prior to detecting communities. Second, the process relies on the assumption that the artificial networks used for training are representative of the real network.

## 5. RESULTS

A SVM with a Gaussian kernel was trained on 8000 100 node networks (described in Section 3) taking the unweighted and weighted mean clustering coefficients as inputs. The “none” class was defined as networks for which the maximum mutual information score was below 0.6. The output classes for the test set (2000 networks) are displayed in Fig. 9. This can be compared to the true class labels in Fig. 8. The overall performance on the test set was 87.1%. A confusion matrix of the test set performance is shown in Table I.

To confirm these results, Fig. 10 shows the mean performance, according to mutual information, when selecting the algorithm class using this classifier. This is compared against the performance of the best weighted algorithm and the best unweighted algorithm.

Table II shows the confusion matrix for the same experiment run on all the networks described in Section 4.3 where the number of nodes in the network equals 1000 (11966 training networks, 17955 test net-

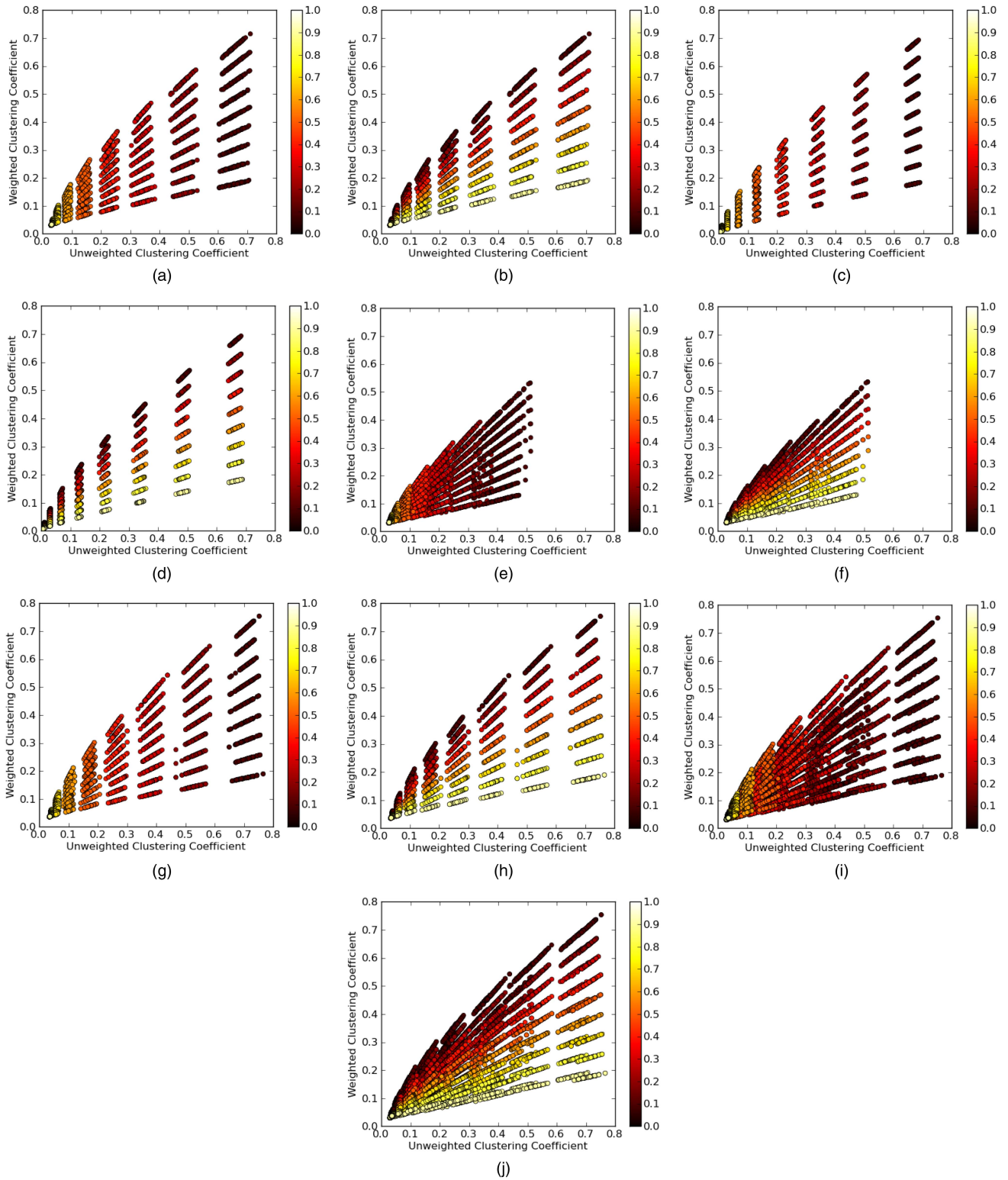


Fig. 7. Scatter plots of the unweighted (x-axis) and weighted (y-axis) mean local clustering coefficient for networks with different parameter settings, the left column shows the value of the topology mixing parameter,  $\mu_t$ . Similarly the right column shows the weight mixing parameter,  $\mu_w$ . (a)  $N = 1000$ ,  $k = 25$ ,  $c_{\min} = 15$ ,  $c_{\max} = 50$  (topology mixing parameter,  $\mu_t$ ). (b)  $N = 1000$ ,  $k = 25$ ,  $c_{\min} = 15$ ,  $c_{\max} = 50$  (weight mixing parameter,  $\mu_w$ ). (c)  $N = 5000$ ,  $k = 25$ ,  $c_{\min} = 15$ ,  $c_{\max} = 50$  (topology mixing parameter,  $\mu_t$ ). (d)  $N = 5000$ ,  $k = 25$ ,  $c_{\min} = 15$ ,  $c_{\max} = 50$  (weight mixing parameter,  $\mu_w$ ). (e)  $N = 1000$ ,  $k = 25$ ,  $c_{\min} = 25$ ,  $c_{\max} = 100$  (topology mixing parameter,  $\mu_t$ ). (f)  $N = 1000$ ,  $k = 25$ ,  $c_{\min} = 25$ ,  $c_{\max} = 100$  (weight mixing parameter,  $\mu_w$ ). (g)  $N = 1000$ ,  $k = 35$ ,  $c_{\min} = 15$ ,  $c_{\max} = 50$  (topology mixing parameter,  $\mu_t$ ). (h)  $N = 1000$ ,  $k = 35$ ,  $c_{\min} = 15$ ,  $c_{\max} = 50$  (weight mixing parameter,  $\mu_w$ ). (i) All networks where  $N = 1000$  (topology mixing parameter,  $\mu_t$ ). (j) All networks where  $N = 1000$  (weight mixing parameter,  $\mu_w$ ).



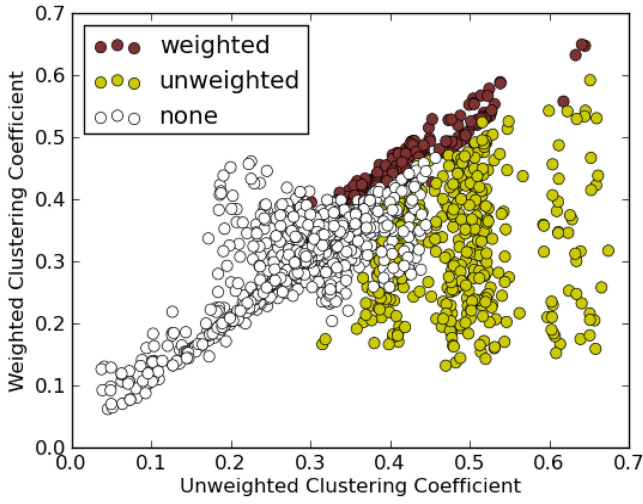


Fig. 8. The true classification of the 100 node networks in the test set.

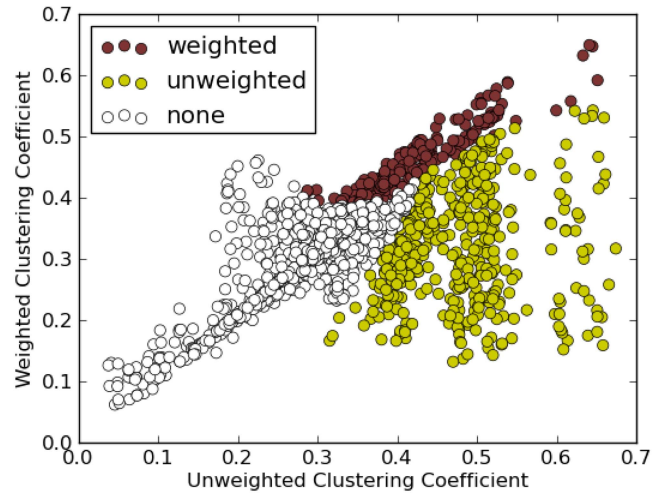
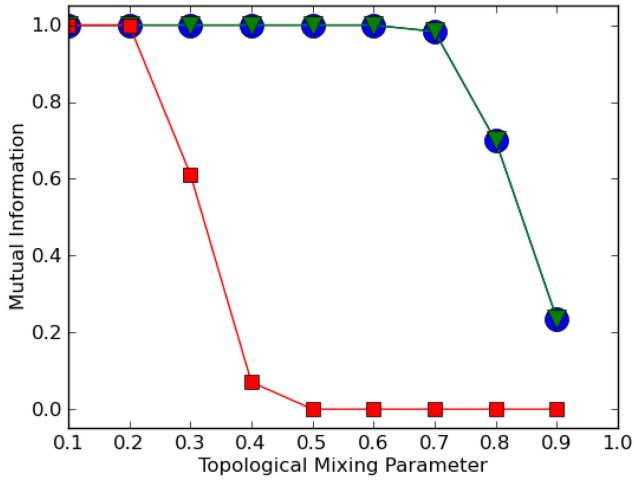
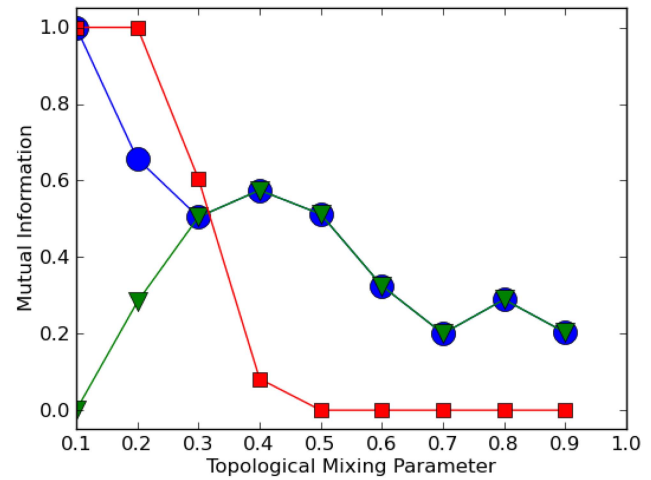


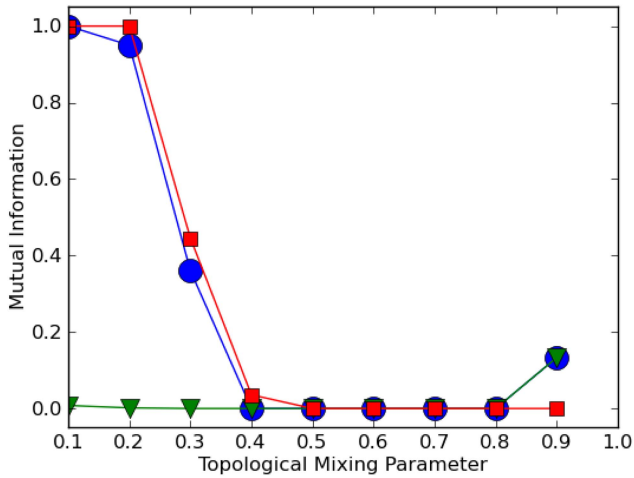
Fig. 9. The predicted classification of the 100 node networks in the test set using a SVM classifier.



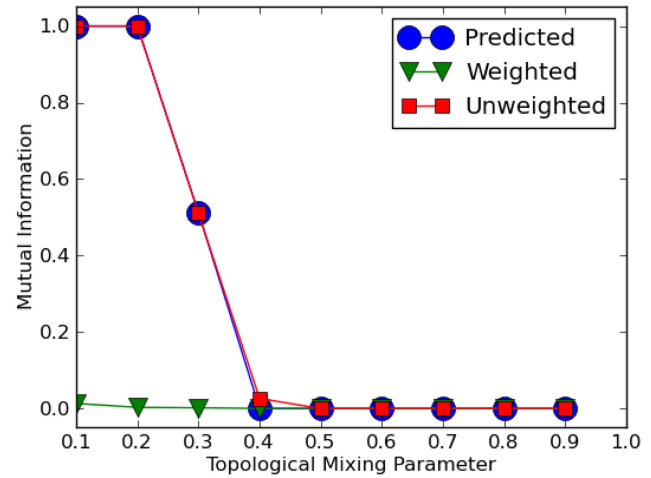
(a)



(b)



(c)



(d)

Fig. 10. Mutual information scores on 100 node networks for when algorithm class is selected by the classifier compared to the individual algorithm classes. Each graph shows the performance (y-axis) as  $\mu_t$  (x-axis) is varied for different  $\mu_w$  values. (a)  $\mu_w = 0.1$ . (b)  $\mu_w = 0.3$ . (c)  $\mu_w = 0.5$ . (d)  $\mu_w = 0.7$ .

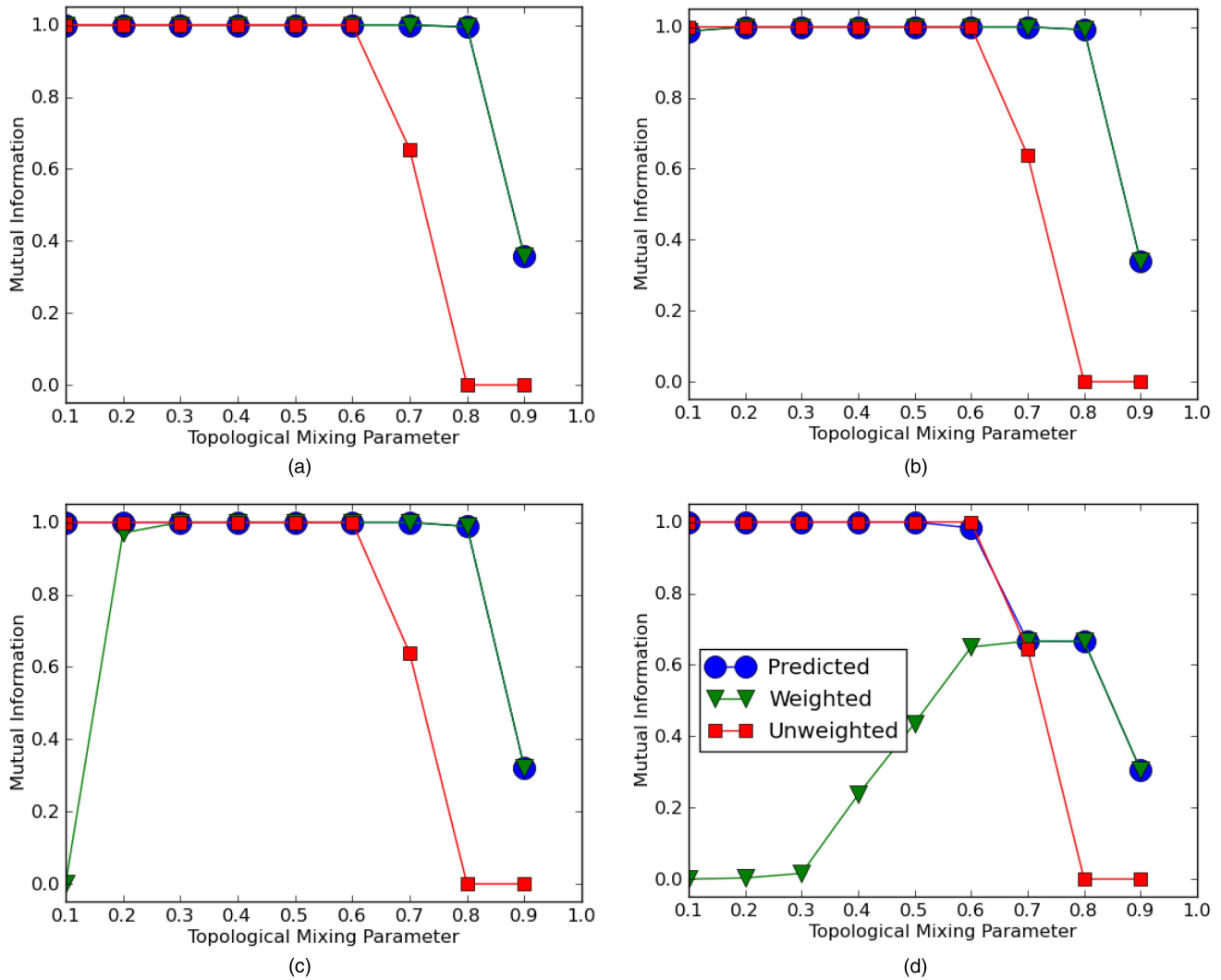


Fig. 11. Mutual information scores on all networks of 1000 nodes for when algorithm class is selected by the classifier compared to the individual algorithm classes. Each graph shows the performance (y-axis) as  $\mu_t$  (x-axis) is varied for different  $\mu_w$  values. (a)  $\mu_w = 0.1$ . (b)  $\mu_w = 0.3$ . (c)  $\mu_w = 0.5$ . (d)  $\mu_w = 0.7$ .

TABLE II  
Classifier Confusion Matrix (1000 nodes)

		Predicted Class		
		Weighted	Unweighted	None
True Class	Weighted	7650	477	467
	Unweighted	85	5237	0
	None	320	17	3702

works). Similarly, Fig. 11 shows the mean performance when selecting the algorithm class using this classifier. From these graphs it can be seen that the classifier is able to select an appropriate class of algorithm such that it can achieve near optimum performance, constrained by the algorithms considered.

From these results it can be seen that even with a simple classifier it is possible to obtain accurate predictions for the best class of community detection algorithm based on properties of the network alone.

## 6. CONCLUSION

To the best of the author's knowledge, no previous work has explored the problem of choosing an appropriate community detection algorithm based on the underlying structural properties. This work has presented community detection algorithms as examples of two classes of algorithm: weighted or unweighted. It is demonstrated that for different types of network and community structure, the class of algorithm has an effect on the performance. It has been shown that for some weighted networks, unweighted algorithms can produce better results than their weighted counterparts. Furthermore it has been shown that it is possible to choose the algorithm class based only on the observed network parameters without prior knowledge of the community structure or assignment.

The algorithm selection demonstrated in this work relies on the training networks (and their communities) to share similar properties as the test network(s). In particular the community size in relation to the size of the

network is shown to have an effect on the relationship between the observable clustering coefficients and the underlying mixing parameters. It is left to future work to identify a means to determine this ratio of community to network sizes.

In addition, the algorithm selection in this work is constrained in terms of the classes of algorithms considered. Future work will reduce these constraints by considering a more comprehensive set of algorithms and classes.

#### ACKNOWLEDGMENTS

This work was undertaken as part of an Engineering Doctorate at the EngD VEIV Centre for Doctoral Training at University College London. UK Patent Pending No.: 1004376.8, March 2010.

#### REFERENCES

- [1] J. Adibi, H. Chalupsky, E. Melz, and A. Valente  
The KOJAK group finder: Connecting the dots via integrated knowledge-based and statistical reasoning.  
*In Proceedings of the national conference on Artificial Intelligence*, 2004, 800–807.
- [2] A. Arenas, L. Danon, A. Diaz-Guilera, P. M. Gleiser, and R. Guimera  
Community analysis in social networks.  
*The European Physical Journal B-Condensed Matter and Complex Systems*, **38**, 2 (2004), 373–380.
- [3] A. Arenas, A. Fernández, and S. Gómez  
Analysis of the structure of complex networks at different resolution levels.  
*New Journal of Physics*, **10** (2008), 053039.
- [4] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre  
Fast unfolding of communities in large networks.  
*Journal of Statistical Mechanics: Theory and Experiment*, (Oct. 2008), P10008.
- [5] A. Clauset, M. E. J. Newman, and C. Moore  
Finding community structure in very large networks.  
*Physical Review E*, **70**, 6 (Dec. 2004), 066111.
- [6] N. Cristianini and J. Shawe-Taylor  
*An introduction to support vector machines*.  
Cambridge University Press, 2000.
- [7] I. Derenyi, G. Palla, and T. Vicsek  
Clique percolation in random networks.  
*Physical Review Letters*, **94**, 16 (Apr. 2005), 160202.
- [8] S. Fortunato  
Community detection in graphs.  
*Physics Reports*, **486**, 3–5 (Feb. 2010), 75–174.
- [9] S. Fortunato and M. Barthélemy  
Resolution limit in community detection.  
*Proceedings of the National Academy of Sciences*, **104**, 1 (2007), 36.
- [10] A. Galstyan and P. R. Cohen  
Identifying covert sub-networks through iterative node classification.  
*In Proceedings of the First International Conference on Intelligence Analysis*, 2005.
- [11] M. Girvan and M. E. J. Newman  
Community structure in social and biological networks.  
*Proceedings of the National Academy of Sciences of the United States of America*, **99**, 12 (June 2002), 7821–7826.
- [12] S. Gregory  
An algorithm to find overlapping community structure in networks.  
*In Knowledge Discovery in Databases: PKDD 2007*, 2007, 91–102.
- [13] S. Gregory  
Finding overlapping communities in networks by label propagation.  
*New Journal of Physics*, **12** (2010), 103018.
- [14] R. Guimerà and L. A. Nunes Amaral  
Functional cartography of complex metabolic networks.  
*Nature*, **433**, 7028 (Feb. 2005), 895–900.
- [15] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral  
Modularity from fluctuations in random graphs and complex networks.  
*Physical Review E*, **70**, 2 (2004), 025101.
- [16] J. Kubica, A. Moore, J. Schneider, and Y. Yang  
Stochastic link and group detection.  
*In Proceedings of the national conference on Artificial Intelligence*, 2002, 798–806.
- [17] A. Lancichinetti and S. Fortunato  
Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities.  
*Physical Review E*, **80**, 1 (2009), 016118.
- [18] A. Lancichinetti and S. Fortunato  
Community detection algorithms: A comparative analysis.  
*Physical Review E*, **80**, 5 (Nov. 2009), 056117.
- [19] A. Lancichinetti, S. Fortunato, and J. Kertész  
Detecting the overlapping and hierarchical community structure of complex networks.  
*New J. Phys.*, **11** (2009), 033015.
- [20] J. Moody  
Race, School integration, and friendship segregation in America.  
*American Journal of Sociology*, **107**, 3 (Nov. 2001), 679–716.
- [21] M. E. J. Newman  
The structure and function of complex networks.  
*SIAM Review*, **45**, 2 (Jan. 2003), 167–256.
- [22] M. E. J. Newman  
Analysis of weighted networks.  
*Physical Review E*, **70**, 5 (Nov. 2004), 056131.
- [23] M. E. J. Newman and M. Girvan  
Finding and evaluating community structure in networks.  
*Physical Review E*, **69**, 2 (Feb. 2004), 026113.
- [24] R. K. Pan and S. Sinha  
Modularity produces small-world networks with dynamical time-scale separation.  
*EPL (Europhysics Letters)*, **85** (Mar. 2009), 68006.
- [25] U. N. Raghavan, R. Albert, and S. Kumara  
Near linear time algorithm to detect community structures in large-scale networks.  
*Physical Review E*, **76**, 3 (2007), 036106.
- [26] J. Reichardt and S. Bornholdt  
Clustering of sparse data via network communities—A prototype study of a large online market.  
*Journal of Statistical Mechanics: Theory and Experiment*, (June 2007), P06016.
- [27] M. Rosvall and C. T. Bergstrom  
Maps of random walks on complex networks reveal community structure.  
*Proceedings of the National Academy of Sciences*, **105**, 4 (Jan. 2008), 1118–1123.
- [28] T. Teitelbaum, P. Balenzuela, P. Cano, and J. M. Buldú  
Community structures and role detection in music networks.  
*Chaos: An Interdisciplinary Journal of Nonlinear Science*, **18** (2008), 043105.
- [29] S. Valverde, G. Theraulaz, J. Gautrais, V. Fourcassie, and R. V. Sole  
Self-organization patterns in wasp and open source communities.  
*Intelligent Systems, IEEE*, **21**, 2 (2006), 36–40.

- [30] K. Wakita and T. Tsurumi  
Finding community structure in mega-scale social networks.  
In *Proceedings of the 16th International Conference on World Wide Web*, Banff, Alberta, Canada, 2007, 1275–1276.
- [31] D. J. Watts and S. H. Strogatz  
Collective dynamics of ‘small-world’ networks.  
*Nature*, **393**, 6684 (June 1998), 440–442.



**Leto Peel** received a M.Eng. in knowledge engineering from the Engineering Mathematics Department of the University of Bristol in 2006.

Since 2006 he has been a research scientist for BAE Systems Advanced Technology Centre in Bristol. He worked in the area of Integrated Vehicle Health Management for 3 years and in 2008 was the winner in the IEEE GOLD category of the Prognostics and Health Management Data Challenge. In 2009 he started an Engineering Doctorate at the Eng.D. VEIV Centre for Doctoral Training at University College London and is supervised by Dr. Simon Julier and Professor John Shawe-Taylor. His research interests include a wide range of machine learning topics, in particular learning from relational data such as social networks.